

Interactive Visualization for the Active Learning Classroom

Dino Schweitzer
Department of Computer Science
United States Air Force Academy, CO 80840
dino.schweitzer@usafa.af.mil

Wayne Brown
Department of Computer Science
United States Air Force Academy, CO 80840
wayne.brown@usafa.af.mil

ABSTRACT

Engaging students in the learning process has been shown to be an effective means for education. Several methods have been proposed to achieve this engagement for computer science and other disciplines. Active learning is one such technique that incorporates interactive classroom activities to reinforce concepts and involve the students. Visualizations of computer science concepts such as algorithm animations can be used for these activities. To be most effective in this environment, they need to be designed and used with active learning in mind. This paper describes the design characteristics of such visualizations, ways of using them in the classroom, and our experience with developing and using visualization tools across different courses in the computer science curriculum.

Categories and Subject Descriptors

K.3.2 [Computer and Information Science Education]: Computer Science education.

General Terms

Design, Experimentation, Human Factors.

Keywords

Active Learning, Collaborative Learning, Visualization.

1. INTRODUCTION

Effective education involves more than simply well-structured formal presentations in the classroom. Research has shown that students learn best when actively engaged in the learning process [8]. Different techniques for engaging students have been proposed over the years with terms such as "active learning", "cooperative learning", "collaborative learning", "problem based learning", "just in time teaching", and "learning-centered focus" [2, 4, 12, 25]. Research results indicate these active approaches are more effective in developing students' ability for higher-order thinking tasks such as analysis, synthesis, and evaluations [19].

Active learning techniques are applicable to all academic disciplines but this discussion will focus on computer science courses. Much active learning occurs in computer science courses due to the project-oriented nature of course assignments, but these activities are typically outside the classroom setting. Specific ideas for applying active learning approaches in the computer science classroom have been described by different authors [5, 14]. Activities such as acting out algorithms, small group problem solving, collaborative activities, games, and classroom worksheets can all be effective active learning techniques for teaching computer science topics [3, 7, 9, 15]. In addition to better education in computer science, active learning has been suggested to reduce the amount of drop-outs in the discipline [24].

This paper proposes the use of interactive visualizations as an effective means to actively engage students in the classroom. Visualization in computer science education is not a new idea and several examples have been described in the literature and posted online (see SIGCSE visualization resources posted at sigcse.org/topics). However, the focus of these visualizations has been on the abstract representations of the concepts and their specific application in the topics they address. Using such tools to support an active learning approach in the classroom requires specific design characteristics that may not be considered when designing general visualizations. Both the design of such tools and their use in the classroom will be addressed based on our experience over several years of developing visualizations for various courses.

2. ACTIVE LEARNING

In its simplest form, active learning has been described as involving students in the classroom in activities other than listening that are meaningful and make them think about what they are doing [4]. Such activities can cover a wide variety of techniques from simple classroom dialoging to complex role-playing. A summary of simple techniques that can be applied in any setting can be found in [17]. Some common characteristics of such activities are:

- Interactive
- Simple to understand
- Short time frame
- Creative and motivational
- Sometimes collaborative
- Relevant

Interactivity is self-evident since the idea of active learning is to engage the student in doing something. The interaction may be with the instructor, with other students, or with some technology (hardware or software). The key is that there is some activity that includes a feedback mechanism that allows a student

to see the results of their actions. It is through this feedback that students gain knowledge about the impact of their actions and, hopefully, about the topic being taught.

An active learning activity must be relatively simple to understand because of limited classroom time. If the activity requires a lot of explanation or is complex to understand, it distracts from the purpose of the activity and uses up valuable classroom time.

Active learning activities should also be relatively short so that they supplement lectures, not replace them. Some researchers have suggested that one of the primary benefits of active learning activities is that they help with students' short attention spans by breaking up a lecture and allowing them to regain focus [19]. Short activities provide an opportunity for the student to experience an engaging "change of pace" that is relevant to the lesson.

Creativity is a desirable quality of an active learning activity because it keeps students interested and engaged. If a student is not motivated to participate in an activity, the level of student involvement will typically be very low. Activities considered "old hat", boring, or "been there, done that" will not keep students interested and can be counter-productive. For example, doing regular repetitive board work can generate boredom or resentment in students. The motivation to participate in an activity may be because the activity is seen as something fun, something different, or something relevant. A motivated participant is more likely to be focused on the activity and gain the desired educational outcome.

Collaboration has been shown to be an effective means of student learning in computer science [10, 13, 15]. Collaborative interaction allows for peer-learning in a supportive non-threatening environment. The element of competition should be minimized or removed. Rather, the focus should be on a group goal-directed activity where the reward is the successful completion of the task. While active learning activities can be effective as either individual or group activities, adding elements of collaboration, where appropriate, can enhance their effectiveness.

Finally, the activity should be viewed by the student as being relevant to the topic being studied. A non-relevant activity, such as doing jumping jacks in the middle of class, may break up the boredom, but does not add to the educational outcome of the lesson. An overly abstract or trivial activity can be confusing or seen as irrelevant. Performance of the activity and its application to the educational concept should be obvious, either through self-evidence or instructor-led discussion. Activities that lack relevancy will not be effective and can reduce enthusiasm for the lesson material.

3. INTERACTIVE CLASSROOM VISUALIZATIONS

The use of visualizations to teach computer science concepts has been applied across many topics for several years. The earliest and most documented examples are in the area of algorithm animations [1, 6, 16]. The advent of flash animation

and web applets accelerated the number of visualizations available to educators. Web searches can find examples for many computer science courses such as algorithms, data structures, graph theory, computer graphics, automata theory, programming languages, computer security, and hardware demonstrations.

Visual demonstrations come in a variety of formats, complexity, level of abstraction, and level of interaction. The least interactive (other than a static image) is a simple animation, such as flash, which only allows the user the ability to control the playback buttons. More sophisticated visualizations allow the user to control elements of the environment such as the input data, environmental parameters, or constraints to see the effects of these changes. Where and how the visualizations are used by the instructor also varies. They can be used for classroom demonstrations, as laboratory exercises, embedded in online learning material, as homework or pre-class exercises, or made available as reference material.

The effectiveness of visualizations as an educational tool has been debated in the literature with mixed results [11]. Looking specifically at algorithm visualizations, Saraiya, et. al., attempted to identify key features of successful visualizations [20]. The results of experiments demonstrated that careful design could measurably increase students' understanding of algorithms.

This paper will focus on visualizations designed specifically for use in the classroom, or ICV's (Interactive Classroom Visualizations). ICV's can be an effective means to promote active learning in the classroom. They should be designed with similar characteristics as other active learning exercises. If they are used correctly in the classroom, they can result in similar educational benefits as other active learning activities.

4. DESIGNING ICV's

At the U.S. Air Force Academy, we have developed and used ICV's in several of our computer science courses including algorithms, data structures, computer graphics, security, cryptography, and introductory computer science [21, 22, 23]. The tools were developed over the years by different instructors. While these tools do not share a single coherent interface or development strategy, our experience is that certain design characteristics promote their use in an active learning fashion in the classroom. Five such design characteristics are described here.

4.1 High Interactivity

The most important design characteristic is a high degree of interactivity. This is similar to Saraiya's finding for effective algorithm animations [20]. To actively involve students, it is necessary to provide a means for them to interact with the ICV in a meaningful way. Ideally, this interaction involves several aspects of the demonstrated concept such as setting parameters, controlling the environment, selecting input data, and stepwise control of the ICV in action. We have observed an important interactive capability is the ability to "back up" the demonstration to review what just happened, or to replay what happens with different conditions. Figure 1 shows a sample interface from a security protocol visualization tool that allows the user to stepwise control the execution of the protocol. The user can

change the protocol at any point in the execution and see the results of the change.

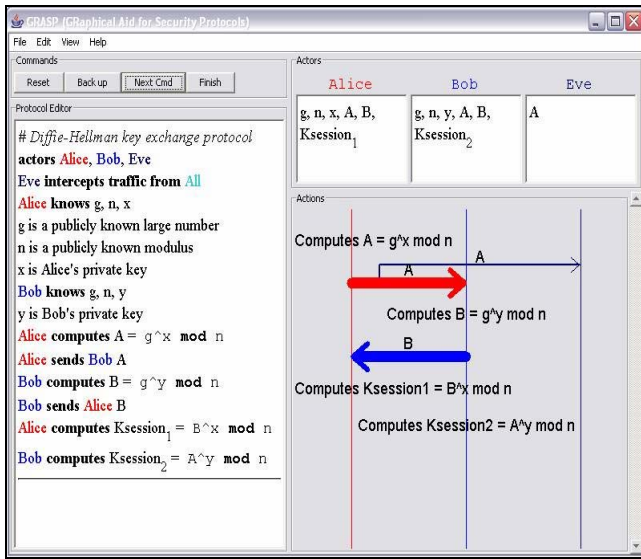


Figure 1. Screen shot of protocol visualization tool.

4.2 Easily Understood Abstraction

By necessity, every ICV contains some level of abstraction of the concepts being illustrated. Visual metaphors represent different data structures, pseudocode summarizes complex algorithms, and motion or graphical indicators can indicate data transfer or actions. The specific abstraction depends on the concept and the way in which the developer chooses to represent objects and actions. This is an area that can involve a great deal of creativity. However, it is important to choose a level of abstraction that can be quickly understood by the student in a classroom setting. Complex representations can take away from the effectiveness of the ICV if the student cannot quickly understand what is happening or is confused by the visual metaphor. Active learning activities should be simple to understand. Their focus must be on the lesson objectives and not on understanding their representation. As an example, the initial design of the protocol visualization tool shown in Figure 1 contained a more complex specification language. After gaining some experience with it, the language was redesigned to be more English-like so students could immediately grasp what was happening. As another example, data transfers are shown with simple arrows in a common timeline format. Color coding was also added to assist the user in associating the transfer originator with the corresponding actor. Other actions are demonstrated with simple representations, such as eavesdropping illustrated with the smaller arrow. Along with easily understood abstractions, well chosen screen designs should be used to minimize clutter and focus attention on the desired actions.

4.3 Simple is Better

A related design guide is to narrow the focus of the ICV to a single or small number of concepts to be demonstrated. It is important in active learning to keep an activity short so that it can be integrated into an overall classroom lesson and be easily

understood. A narrow focus can be achieved partially through the design of the ICV, as well as how it is used in the classroom. Using an ICV with complex interactions and many options can dilute the students' focus. As an example, Figure 2 shows one design of several ICV's that were created for teaching ciphers. Each individual ICV focuses on a single cipher algorithm (a Vigenere cipher in this example). An additional design feature in this series of ICV's is the use of tabs to select which characteristic or cipher attack to demonstrate. This allows the instructor to demonstrate and students to interact with a single concept regarding the cipher without the visual clutter or confusion of currently non-relevant material. In this example, the instructor can use different tabs, such as the "Key Length" tab, at different points in the lecture as a separate active learning activity.

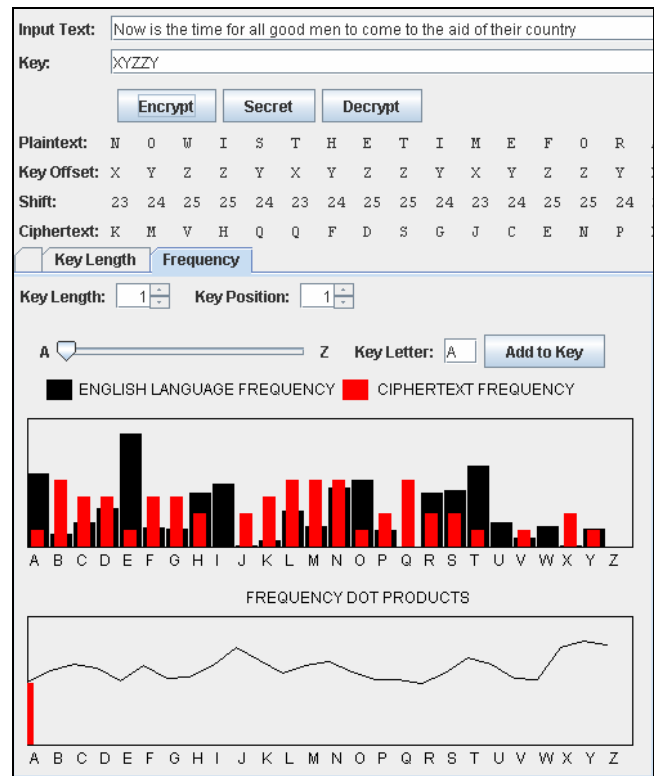


Figure 2. Cipher visualization tool.

4.4 Relevancy

The level of abstraction and concept focus of the ICV can help relevancy by removing extraneous or confusing aspects of a topic. However, one of the primary challenges in relevancy is the selection of what concepts to represent with ICV's in the first place. If ICV's are all inclusive, such as all algorithms taught in an algorithms course, there is a danger of the "ICV for the day" to become boring and less meaningful. Consideration should be given to concepts that meet one or more of the following qualities: hard for students to understand, representative of a class of concepts, can be translated into a meaningful visualization, have a particular interest to students, or support the structure of the course. For example, when developing the aforementioned cipher ICV's, the ciphers were selected based on historical relevance, a simple cipher to demonstrate the tool, an example of

a modern cipher, and ciphers with well-known problems or attacks.

4.5 Robust User Interface

A lesson learned from our experience in using ICV's in the classroom is to keep the interface simple and minimize opportunities for user error. For example, the protocol ICV shown in Figure 1 allows for protocol specifications to be pre-defined and loaded from a file. This minimizes error-prone keyboard input. The cipher applets make extensive use of buttons, sliders, and GUI components. Robustness from incorrect user input is also important, especially when being operated by inexperienced students. Tool crashing during demonstration is not a positive experience for the student or instructor.

5. CLASSROOM USE

In addition to design characteristics, effective use of ICV's in an active learning environment requires consideration of how the tool is presented to students and used to enhance the lecture. If the tool is simply demonstrated by the instructor, then it becomes an extension of a lecture and does not engage the student other than being a different presentation approach. A more interactive approach by having the students operate the tool, either directly or through the instructor, provides a hands-on activity more in line with active learning. Some examples of these approaches follow.

The easiest way to involve students as more than spectators is to set up "what if" situations and have them predict the results. For example, the user can change the key in a cipher algorithm and challenge students to predict the effect on the resulting ciphertext. Or, asking the students to predict the effects of a replay attack in a security protocol. The design consideration of interactivity is important for this approach to be able to change conditions and stepwise stop and back up execution. Students can be engaged in this individually, in groups, or as an entire class.

Another ICV classroom approach is to direct a goal-oriented activity, once again individually or in a small group format. For example, consider the computer graphics ICV for color theory shown in Figure 3. The typical demonstration allows the user to combine red, green, and blue components and displays the resulting mixed color. It is non-intuitive in additive RGB space how much of which color to add to match a desired result. Having a student operate the tool while other students are providing direction and encouragement can be both entertaining and instructional. Students can also operate the ICV as a small group activity promoting collaborative active learning.

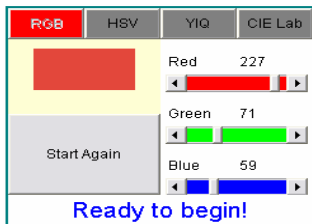


Figure 3. Interactive color applet from www.cs.rit.edu.

6. OUR EXPERIENCE

Our experience with the use of ICV's at the U.S. Air Force Academy is that they are popular with the students and add a "gee whiz" component to the traditional lecture. Students are motivated to interact with them and willingly participate in classroom activities involving the tools. We provide access to the tools outside of class to allow students the opportunity to "play" with them. In course evaluations, students rate the visualizations as a fun and productive part of the course. While we have not conducted formal studies as to the ICV's effect on student learning, the positive student reaction and the evidence in educational literature on the effects of active learning approaches have convinced us that this is a productive approach.

Designing ICV's for active learning has been an educational process for us as we have gained experience with their design and use. One tendency we have learned to avoid is trying to put too much into a single visualization (creeping featurism). Keeping them short and focused is important for effective classroom use. Another lesson learned is to keep the interface as mouse-driven as possible and avoid the keyboard. Extensive typing by the user (instructor or student) in a darkened classroom in front of an audience can be a stressful experience. When visualizing algorithms, we found it better to use fewer steps at a higher level of pseudocode as shown in Figure 4. Highlighting which step of the algorithm is executing and being able to undo steps and their effects is important for student interaction. Finally, we discovered that judging students' reactions and how quickly they will comprehend an abstraction is difficult a priori. The best measure is to try something in the classroom, gauge the reaction, and be willing to change the ICV accordingly.

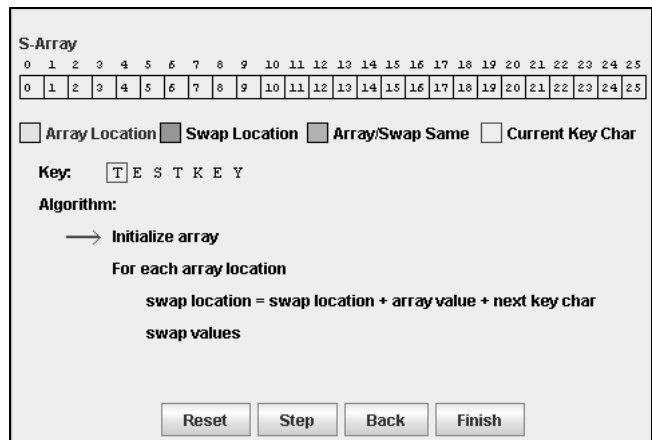


Figure 4. Sample use of pseudocode in ICV.

We are continually developing additional ICV's for other courses and concepts in the major. We have even used ICV tool development as a course project in our software engineering sequence as well as independent study projects. Students enjoy the challenge of developing them and having their work demonstrated in front of their peers. ICV's will continue to be used as a tool in our inventory of active learning activities.

7. REFERENCES

- [1] Baecker, R. Sorting out Sorting. 16mm color sound film, 25 minutes, 1981.
- [2] Bailey, T. and Forbes, J. 2005. Just-in-time teaching for CS0. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education* (St. Louis, Missouri, USA, February 23 - 27, 2005). SIGCSE '05. pp. 366-370.
- [3] Begel, A., Garcia, D., and Wolfman, S. Kinesthetic learning in the classroom. In *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education* (Norfolk, Virginia, USA, March 03 - 07, 2004). SIGCSE '04. pp. 183-184.
- [4] Bonwell, C., and Eison, J. Active Learning: Creating Excitement in the Classroom. ASHE-ERIC Higher Education Report 1, 1991.
- [5] Briggs, T. 2005. Techniques for active learning in CS courses. *J. Comput. Small Coll.* 21, 2 (Dec. 2005), pp. 156-165.
- [6] Brown, M. Perspectives on algorithm animation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Washington, D.C., United States, May 15 - 19, 1988). J. J. O'Hare, Ed. CHI '88. pp. 33-38.
- [7] Budd, T. An active learning approach to teaching the data structures course. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education* (Houston, Texas, USA, March 03 - 05, 2006). SIGCSE '06, pp. 143-147.
- [8] Chickering, A. and Gamson, Z. Seven principles of good practice in undergraduate education. *AAHE Bulletin*, 39, pp. 3-7, 1987.
- [9] Fleury, A. Acting out algorithms: how and why it works. *Proceedings of the Fourth Annual Consortium for Computing in Small Colleges Midwestern Conference* published as *The Journal of Computing in Small Colleges*, November 1997, 13 (2), pp. 83-90.
- [10] Gonzalez, G. 2006. A systematic approach to active and cooperative learning in CS1 and its effects on CS2. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education* (Houston, Texas, USA, March 03 - 05, 2006). SIGCSE '06, pp. 133-137.
- [11] Hundhausen, C., Douglas, S., and Stasko, J. A meta-study of algorithm visualization effectiveness. *Journal of Visual Languages and Computing* 13, pp. 259-290, 2002.
- [12] Johnson, D., Johnson, R., and Holubec, E. Advanced Cooperative Learning. Interaction Book Co., Edina, MN, 1992.
- [13] Ludi, S., Natarajan, S., and Reichlmayr, T. An introductory software engineering course that facilitates active learning. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education* (St. Louis, Missouri, USA, February 23 - 27, 2005). SIGCSE '05, pp. 302-306.
- [14] McConnell, J. Active learning and its use in computer science. In *Proceedings of the 1st Conference on integrating Technology into Computer Science Education* (Barcelona, Spain, June 02 - 06, 1996). ITiCSE '96, pp. 52-54.
- [15] McKinney, D. and Denton, L. Developing collaborative skills early in the CS curriculum in a laboratory environment. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education* (Houston, Texas, USA, March 03 - 05, 2006). SIGCSE '06. pp. 138-142.
- [16] Naps, T., Rößling, G., Almstrum, V., Dann, W., Fleischer, R., Hundhausen, C., Korhonen, A., Malmi, L., McNally, M., Rodger, S., and Velázquez-Iturbide, J. Exploring the role of visualization and engagement in computer science education. In *Working Group Reports From ITiCSE on innovation and Technology in Computer Science Education* (Aarhus, Denmark, June 24 - 28, 2002). ITiCSE-WGR '02. pp. 131-152.
- [17] Paulson, D. and Faust, J. Active learning in the college classroom. *Journal on Excellence in College Teaching*, vol. 9 (2), pp. 3-24, 1998.
- [18] Pollard, S. and Duvall, R. Everything I needed to know about teaching I learned in kindergarten: bringing elementary education techniques to undergraduate computer science classes. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education* (Houston, Texas, USA, March 03 - 05, 2006). SIGCSE '06. pp. 224-228.
- [19] Prince, M. Does active learning work? A review of the research. *Journal of Engineering Education*, 93:3, pp. 223-231, 2004.
- [20] Saraiya, P., Shaffer, C. A., McCrickard, D. S., and North, C. Effective features of algorithm visualizations. In *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education* (Norfolk, Virginia, USA, March 03 - 07, 2004). SIGCSE '04, pp. 382-386.
- [21] Schweitzer, D. Designing interactive visualization tools for the graphics classroom, ACM SIGCSE Bulletin, v.24 n.1, p.299-303, March 1992.
- [22] Schweitzer, D., Baird, L., Collins, M., Brown, W., and Sherman, M. GRASP: a visualization tool for teaching security protocols. In *Proceedings from the Tenth Colloquium for Information Systems Security Education*. Adelphi, MD, June 5-8, 2006.
- [23] Schweitzer, D. and Baird, L. The design and use of interactive visualization applets for teaching ciphers. In *Proceedings of the 7th IEEE Workshop on Information Assurance*. U.S. Military Academy, West Point, NY, June 21-23, 2006.
- [24] Seymour, E. and Hewitt, N. *Talking about leaving*. Westview Press, Boulder, CO., 1997.
- [25] Sutherland T. and Bonwell, C. Using active learning in college classes: a range of options for faculty. *New Directions for Teaching and Learning*, no. 67. San Francisco, Jossey-Bass, 1996.