The Incorporation of Algorithm Visualization in Computer Science Hypertextbooks

Guido Rößling
Department of Computer Science
Darmstadt University of Technology
D-64289 Darmstadt, Germany
roessling@acm.org

Thomas L. Naps
Department of Computer Science
University of Wisconsin Oshkosh
Oshkosh, WI 54901
naps@uwosh.edu

ABSTRACT

An increasing body of evidence suggests algorithm visualization (AV) is effective only in conjunction with other techniques that force a degree of user interaction beyond the mere "watching" of an algorithm [3]. Hence, whereas much previous work on AV has tended to focus on the visualizations themselves, the real key to the success of AV may be the techniques we use to launch the visualizations from other materials that regard the visualization as a resource in an arsenal of instructional aids. One such "launching technique" is a *hypertextbook* [1], which we broadly define as a Web-accessible textbook with hyperlinks to a variety of other resources. This working group will specifically explore the most advantageous ways of incorporating AV as one of those "other resources" that is employed by a hypertextbook.

1. BACKGROUND

An increasing body of evidence suggests algorithm visualization (AV) is effective only in conjunction with other techniques that force a degree of user interaction beyond the mere "watching" of an algorithm [3]. These techniques include having learners answer questions about the visualization, having learners experiment with providing specific kinds of data sets to the algorithm being visualized, and having learners design the visualization themselves. The tendency of AV researchers has been to focus on the visualizations themselves. However, if visualizations are only effective when combined with these other proven effective techniques, then it makes sense for computer science educators who want to use AV (as opposed to develop AV systems) to explore integrating AV into their courses in a fashion that naturally demands their students use AV in conjunction with these techniques. If this is not done, the danger is that students will continue to feel that AV is just another "add-on" – something to try if they feel inclined to do so, but not something that is necessary for their success in the course.

One way of encouraging students to regard AV as essential for learning and ensure that they use it in a directed way is to incorporate it into their textbook for the course. In [1], Boroni, Goosey, Grinder, and Ross introduced the notion of a *hypertextbook* for computer science instruction:

> In their most rudimentary incarnations, hypertextbooks may simply make effective use of hyperlinks that allow the reader to branch to related portions of the text by clicking a mouse button on a section of highlighted text. [...] An active learning hypertextbook [...] will additionally incorporate interactive software modules [...] that actively engage students in the learning experience.

With regard to hypertextbooks in computer science, we feel that AV systems are the natural choice to be the "interactive software module" that Boroni, Goosey, Grinder, and Ross envision. Yet, other than the effort by Grinder, Kim, Lutey, Ross, and Walsh to develop such a hypertextbook for a theory course [2], little progress has been made in the authoring of such hypertextbooks augmented with AV modules.

2. GOALS, METHODOLOGY, AND ACTIVITIES

The goals of this working group (WG) are twofold:

1. Explore in general why hypertextbooks have not been widely adopted as a means for delivering course content in computer science.
2. Explore specifically how AV systems can be incorporated in a natural way with the material that is presented to students in a hypertextbook.

With regard to the first goal, the group will address such questions as:

- What are methods for publishing hypertextbooks? The Web is the obvious medium, but are there others?
- The development of such hypertextbooks is, in many ways, more daunting for authors than the development of a traditional textbook because it entails integrating reliable software modules into learning material. As such, authoring a complete hypertextbook may well require collaborative efforts among relatively

large groups of authors and software developers. How can such a collaboration be carried out effectively, e.g. using a Wiki?

- What added benefit or other incentive is needed for teachers and students to adopt a hypertextbook in addition to, or even as a replacement of, "established literature"?
- Should the authoring participation in a hypertextbook be "open" (as in Wikipaedia), "closed" (as for textbooks), or "semi-open", e.g. for registered teachers? What implications arise from these models for adoption chances, accuracy, and ease of use?
- Is it realistic to expect that the authors of hypertextbooks will get the same kind of "pay-back" that authors of traditional textbooks have received from publishers? What will it take to get publishers to buy into the notion of a hypertextbook?

For AV software developers, guidelines in regard to the second goal above will be particularly important. For example:

- The obvious way to incorporate software modules into hypertextbooks published on the Web is to use Java applets. But is this only way? The best way? One company with which one of the co-chairs is familiar has a company-wide policy to "never put an applet between the customer and the buy button" precisely because of the notorious unreliability of Java applets and their reliance upon the idiosyncrasies of Web browsers.
- What are alternatives to applets for the development of such hypertextbook-launchable AV? What are the advantages and disadvantages of these alternatives?
- To what degree will AV software integrated into such hypertextbooks also have to know how to "talk to" course management software, for example, databases in which the results of student usage of the system is recorded?

In electronic communication before the working group convenes in Bologna, we will collect a variety of views from WG members on the answers to these questions and also encourage members to formulate other relevant questions that the group may want to address.

During ITiCSE 2006, the group will develop specifications for the layout, design, structure, etc., of hypertextbooks that actively incorporate AV. In one sense, our report will hopefully become a "blueprint" for authors and AV developers who wish to collaborate on hypertextbook efforts in the future.

After ITiCSE 2006, we hope that many members of the working group will start such collaborations in hypertextbook authoring and then report on these efforts for ITiCSE 2007.

3. QUALIFICATIONS OF THE CO-CHAIRS

Guido Rößling received the Diploma in Computer Science from the Darmstadt University of Technology, Germany, in 1996. From 1996 to 2001, he worked as a research assistant at the University of Siegen, Germany. He finished his Ph.D. thesis on AV system design in 2002. In November 2001, he joined the Darmstadt University of Technology as a research assistant for e-learning applications.

Since 1998, he has developed the extensible AV system Animal that is now also used in Naps' JHAVÉ system. He has published his research on e-learning applications since 2000. This includes several conference papers and journal articles on AV. He was a member of the program chair for the 2002, 2004, and 2006 Program Visualization Workshop, held in conjunction with ITiCSE 2002, 2004, and 2006.

Tom Naps received the PhD in Mathematical Logic from the University of Notre Dame in 1975. Since then he has taught a broad range of mathematics and computer science courses, first in the University of Wisconsin Center System (1975-81), then at Lawrence University (1981-2001), and now at the University of Wisconsin - Oshkosh.

Since 1987 he has pursued AV both from the perspective of an instructor who wants to design visualizations of particular algorithms to help his students and as the developer of the GAIGS and JHAVÉ AV systems. Naps has written twelve papers in the area of AV, conducted workshops on AV under the NSF's Undergraduate Faculty Enhancement Program (1991), conducted a workshop on AV at the 1992 ACM SIGCSE Technical Symposium, conducted a tutorial on Java-based AV at the 2000 ITiCSE conference, and co-chaired previous international working groups on visualization at recent ITiCSE conferences. Over sixty faculty members at other institutions have used his AV systems. In developing GAIGS and JHAVÉ, he has worked with over twenty undergraduate research assistants. He has collaborated with John Stasko of Georgia Tech and Guido Rößling of the Darmstadt University of Technology to incorporate their scripting languages (Samba and AnimalScript respectively) into the JHAVÉ environment. He is currently working with Scott Grissom (Grand Valley State University) and Myles McNally (Alma College) under a three-year National Science Foundation grant to develop instructional materials to support AV.

4. POTENTIAL PARTICIPANTS

All of the following have been contacted to provide feedback on our proposal. All have indicated they would consider participating in the group if their time and financial constraints allow them to do so:

- Scott Grissom, Grand Valley State University, Allendale, MI
- Mark Hall, University of Wisconsin Marathon County, WI
- Chris Hundhausen, Washington State University, Pullman, WA
- Duane Jarc, University of Maryland University College, MD
- Ville Karavirta, University of Joensuu, Finland
- Andreas Kerren, University of Kaiserslautern, Germany
- Ari Korhonen, Helsinki University of Technology, Helsinki, Finland
- Chuck Leska, Randolph-Macon College, VI

- J. Ben Schafer, University of Northern Iowa, IA
- J. Ángel Velázquez Iturbide, Universidad Rey Juan Carlos, Madrid, Spain

The following have been contacted to provide feedback on our proposal. They consider participating in the online group discussions, but their time constraints do not allow them to be at the meeting in Bologna:

- Stephan Diehl, KU Eichstätt, Germany
- Rockford Ross, University of Montana, MN

5. REFERENCES

[1] C. M. Boroni, F. W. Goosey, M. T. Grinder, and R. J. Ross. Engaging Students with Active Learning Resources: Hypertextbooks for the Web. In *Proceedings of the* 32^{nd} *ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE 2001), Charlotte, North Carolina*, pages 65–69. ACM Press, New York, 2001.

[2] M. T. Grinder, S. B. Kim, T. L. Lutey, R. J. Ross, and K. F. Walsh. Loving to Learn Theory: Active Learning Modules for the Theory of Computing. In *Proceedings of the* 33^{rd} *ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE 2002), Cincinnaty, Kentucky*, pages 371–375. ACM Press, New York, 2002.

[3] T. L. Naps, G. Rößling, V. Almstrum, W. Dann, R. Fleischer, C. Hundhausen, A. Korhonen, L. Malmi, M. McNally, S. Rodger, and J. Á. Velázquez-Iturbide. Exploring the Role of Visualization and Engagement in Computer Science Education. *ACM SIGCSE Bulletin*, 35(2):131–152, June 2003.