Merging Interactive Visualizations with Hypertextbooks and Course Management

Guido Rößling
TU Darmstadt, Germany
roessling@acm.org

Thomas Naps
U Wisconsin Oshkosh, USA
naps@uwosh.edu

Mark S. Hall
U Wisconsin Marathon County, USA
Mark.Hall@uwc.edu

Ville Karavirta
Helsinki University of Technology, Finland
vkaravir@cs.hut.fi

Andreas Kerren
U Kaiserslautern, Germany
kerren@acm.org

Charles Leska
Randolph-Macon College USA
cleska@rmc.edu

Andrés Moreno
University of Joensuu, Finland
amoreno@cs.joensuu.fi

Rainer Oechsle
FH Trier, Germany
oechsle@fh-trier.de

Susan H. Rodger
Duke University, USA
rodger@cs.duke.edu

Jaime Urquiza-Fuentes
Universidad Rey Juan Carlos Spain
jaime.urquiza@urjc.es

J. Ángel Velázquez-Iturbide
Universidad Rey Juan Carlos Spain
angel.velazquez@urjc.es

ABSTRACT

As a report of a working group at ITiCSE 2006, this paper provides a vision of how visualizations and the software that generates them may be integrated into hypertextbooks and course management systems. This integration generates a unique synergy that we call a Visualization-based Computer Science Hypertextbook (VizCoSH). By borrowing features of both traditional hypertextbooks and course management systems, VizCoSHs become delivery platforms that address some of the reasons why visualizations have failed to find widespread use in education.

The heart of the paper describes these features and explains, from both a student and teacher perspective, how each feature adds educational value to a visualization. In some cases, this value focuses on pedagogical issues, taking advantage of known strategies for making visualizations more engaging and effective. In other cases, the emphasis is on making it easier for teachers to use visualizations. A set of possible use scenarios and approaches for increasing interest in adopting a VizCoSH are also presented.

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer & Information Science Education - *Computer Science Education*

General Terms

Algorithms

Keywords

Visualization, Animation, Pedagogy, Hypertextbooks

1. INTRODUCTION

The term *hypertextbook* has no fixed standard definition. This lack of a standard definition by no means reflects a lack of literature that has been written about the topic – features of a hypertextbook have been described in a variety of contexts and disciplines. Section 2.1 references some of these hypertextbook definitions.

This wealth of varied material that has already been written on hypertextbooks places it well beyond the scope and desire of this working group to try to create a standard definition that will satisfy a large number of people, from a variety of disciplines, who are already very knowledgeable about hypertextbooks from a broad perspective. Instead our perspective is that of the designers of pedagogic visualization systems who have been disappointed that our systems have not found widespread use despite:

1. The strongly held belief among CS teachers that such systems help learners [44], and

2. The growing amount of evidence showing such systems are indeed effective when used wisely [44].

This perspective gives us the latitude to define the features of "our ideal hypertextbook" in a way that will have the best chance

of actually affecting the way visualization is used in CS education. In that sense, the approach we will take is admittedly limited and highly focused. We will use the term Visualization-based Computer Science Hypertextbook (VizCoSH) to distinguish this focused approach from broader descriptions of hypertextbooks in general. Nonetheless, we will certainly rely on more general discussions of hypertextbooks in the literature to "pick and choose" from those features that are most advantageous to use in VizCoSHs.

Minimally, a VizCoSH must:

1. Be text-based – otherwise, it is not a textbook. This text co-ordinates and sets the stage for the student's use of the visualizations. Such text can be incorporated into the VizCoSH at a conceptual level before the student is ever led to an actual visualization, or it can be text that explains to the student how to interpret a particular visualization, that is, how to "make sense of the pictures the student sees" (significance of color usage, etc.);

2. Have significant structure, not just be disconnected text;

3. Exhibit some features of textbooks, such as a table of con-tents, chapters, and index. These features enable students to use it as a textbook for an entire course or at least a particular module in a course;

4. Offer navigations based on links that lead to visualizations. Moreover, unless such links lead to visualizations that offer interactive content and satisfy at least a subset of features that will be discussed in this paper, the links will not take significant advantage of visualization systems and hence are not pertinent to this group's perspective.

Beyond these minimal criteria, features of a VizCoSH may also include tools that are more typically thought of as a course management system. These include the ability to log student use of the VizCoSH resources, record student responses to assessment questions and activities, and allow students to post their own additions and annotations to the VizCoSH materials.

Our rationale for the desired features of VizCoSHs will be broken down into two perspectives – advantages offered to students who learn from them and advantages offered to teachers who employ VizCoSHs as a teaching resource.

Advantages offered to students: We envision a VizCoSH as being at its core a textbook resource for the student. It possesses all the features of a paper textbook (chapters, sections, exercises, indices, table of content, …) and is additionally available through links in the normal fashion and, ideally, additional links that enable different paths through the material. In addition, our view places a high value on incorporating interactive visualizations into the heart of this text. The types of interactivity that can be promoted by VizCoSHs will be discussed in Section 2 and include a variety of engagement strategies whose effectiveness has been documented in prior literature.

Advantages offered to teachers: From the teacher's perspective, VizCoSHs should be easy to install, easy for their students to learn to use, offer consistency of presentation, enable customizations, and have some type of grading support. Some teachers will hand out the VizCoSH as a supplement and will not use it during lecture, thus wanting students to be able to install and figure out how to use it on their own. Teachers will want the presentation of the material consistent throughout the VizCoSH (and consistent with the course textbook if the VizCoSH is a supplement). Some teachers will want a VizCoSH that is modifiable, so they can customize it for their own course, while others will want a VizCoSH that is ready to go with selectable examples. Other teachers will want a VizCoSH that includes features of a course management system, such as tracking student usage. Such monitoring may take the form of logging usage statistics for data collection purposes or recording results of student engagement activities for assessment purposes.

2. RELATED WORK

2.1 Variation of Hypertextbooks

A general review of the literature defines a hypertextbook as "a regular textbook translated to HTML and delivered on the Web with lots of hyperlinks" [7], with publishers desiring tools to automate the translation of textbooks to hypertextbooks [12]. A unique view of a hypertextbook emphasizes the use of a database and a search engine to retrieve URLs [6]. A third viewpoint that best corresponds to our vision of a hypertextbook is one that greatly extends the capability of a textbook and "allow[s] for different learning paths through the material for different learning needs, an abundance of pictures and illustrations, video clips where helpful, audio, and – most importantly – interactive, active learning visualizations of key concepts." [50]

This last viewpoint is being implemented by Ross both as a computer science hypertextbook, *Snapshots of the Theory of Computing* [49], and as a biological engineering hypertextbook, Biofilms [13]. Biofilms is a new discipline that needs multimedia movie clips as part of the course content, which makes it a natural fit for a hypertextbook. Ross has taken his viewpoint on hypertextbooks and introduced a course on incorporating animation and visualization into a hypertextbook [48], as shown in Figures 1 and 2.

Both of Ross' hypertextbooks are works in progress. His theory hypertextbook started over twelve years ago, showing the intense effort needed to produce a hypertextbook. Another example of in-tense effort is a Physics hypertextbook under construction by G. Elert that will have "seven books, each book will contain about seven chapters, and each chapter will contain about seven sections. (More or less.)" [17]

2.2 Engagement Taxonomy

Students will benefit most by using a VizCoSH. Current visualization systems can provide several levels of student engagement as listed by Diehl and Kerren [14] in their *levels-of-explorations* approach, published 2001, and by Naps et al. [44] in their *engagement taxonomy* based in part on Bloom's taxonomy [3, 43], published 2003. As the latter concept partly subsumes the different exploration levels of the first referenced paper, we will describe the engagement taxonomy in more detail. Six main forms of engagement/interaction were identified–no viewing, viewing, responding, changing, constructing, and presenting. *No viewing* can be considered as the category for traditional hypertextbooks where no visualization technologies are used. In this section we discuss how visualization and its more developed forms of engagement offer an improvement over the existing CS hypertextbooks counterparts. Moreover, visualization brings into the picture new forms of engagement that have not been included previously in hypertextbooks.

Viewing. This is the basic engagement feature that every visualization tool provides as it only considers the playback methods for visualizations. Classical hypertextbooks provide static images, and our approach will certainly

include them. Additionally, animated visualizations will be more suitable for the dynamic processes common in CS, *e.g.*, program execution or distributed protocols.

Responding. New ways to assess student knowledge appear when using visualizations, *e.g.*, predicting the next step or frame.

As well, tests carried out in classical hypertextbooks can be extended with visualizations. This way students will answer the questions within a supporting visual context. For example, detailed animations could be used as explanations of why a certain answer was correct or incorrect.

Changing. Visualizations usually include the means to modify the animation, either by setting its graphical parameters or, better, by letting the student modify the algorithm parameters, *e.g.*, keys to be sorted by an algorithm. The latter will produce a completely new visualization. Classic hypertextbooks have included applets that were limited to the production of a static diagram given the parameters that define the diagram. The benefits of animations are clearer when learning dynamic concepts.

These three engagement levels can be seen in classic hypertextbooks as well, but visualizations can bring these forms of interaction to a different level. Students can now check their knowledge and mental models by interacting with the material of the hypertextbook. Thus, it is not only providing animated content but also reacting and supporting students better than classic hypertextbooks when they encounter conflicts or doubts processing the textual information.

The three engagement levels mentioned before can be seamlessly integrated into the hypertextbook. Details on achieving this will be presented in Section 6. The next two engagement levels offer more interesting challenges:

Constructing. By allowing students to construct their own visualizations of an algorithm, visualization tools promote a deeper engagement of the algorithm [27]. Hundhausen and Douglas [28] identified two ways of constructing visualizations, either by adding graphical commands to an (existing) algorithm [11], or using an animation editor [51]. Thus, the visualizations will represent the point of view of the student.

Presenting. Finally, either the self-constructed animations or animations obtained from other sources, can be presented by the students to foster discussion and feedback. VizCoSHs can support presenting by letting students add their own material, especially visualizations, to the repository of VizCoSH materials.

Constructing requires us to move out of the VizCoSH and employ the already developed visualization tools as the basis for students to develop their visualizations. A VizCoSH can support students in this endeavor by providing information on using such tools. Students, supposedly, have already been working with the animations produced by these tools, and they have a better idea about what they can achieve with them. Moreover, the presenting activity can heighten students' motivation and engage them in actively constructing the visualization [46].

2.3 Concerns Impacting Use of Visualizations

A pre-conference survey for the working group on visualization at ITiCSE 2002 [44] asked teachers why they and their colleagues were reluctant or unable to use dynamic visualizations. We give a summary of the twenty-nine respondents. The main reasons teachers were reluctant to use dynamic visualizations dealt with time: time required to search for good examples in the literature and on the web, time to learn the new tools, and time to develop the visualizations. Additional reasons given in the survey include:

- lack of effective and reliable visual development tools for both teachers and students

- time to adapt visualizations into the course curriculum and to transition them into the classroom

- concern that they are unsure about how to integrate technology into a course and into the classroom

- time and support for installation of visualization software

- concern about classroom facilities that support multimedia presentations (some classrooms have lights that are either on or off and there is no dimming feature.)

- lack of empirical evidence that shows the effectiveness and benefits of visualizations in learning

- concern that visualizations themselves may not adequately show the important details in the concept being displayed

- concern that students are too passive.

When regarding these complaints in more detail, they seem to indicate a prevalent attitude: using visualizations increases the (usually already high) workload for educators, while no profound improvements in learning are expected. This may also explain the common concerns about time: if visualizations were proven to be effective, they would be more likely to be given a high priority. This would probably also cause the timing concerns to be seen as less critical. Arguably, the same combination of perceived complications and concerns about effectiveness apply to most other novel teaching approaches that have not yet "caught on".

VizCoSHs present opportunities to address these concerns. No external searching for visualizations is required, since they would be incorporated directly into the VizCoSH. Time for learning and developing visualizations is reduced due to consistency within the visualizations themselves and the seamless integration of the visualization with the VizCoSH. Visualizations will be easy to introduce into a course, as they are part of the VizCoSH, which is viewed by students as an essential resource (e.g., the VizCoSH could well be the only textbook resource for the course). These improvements would promote adoption by more teachers, which would lead to the increase of empirical studies that correlate these visualizations with learning. Additionally, the incorporation of visualizations into the VizCoSH makes their use "natural", and can further help in overcoming some of the concerns stated above.

3. BEST PRACTICES

3.1 Overview of Didactical Aspects in VizCoSH

Our vision of a VizCoSH is to take the best practices of hypertextbooks and the best practices of visualization systems to create an all-encompassing learning tool. These best practices of hypertextbooks are the following:

- Basic sequential structure of paper textbooks including table of contents and index (see Figure 1, taken from [49])
- Additional navigational structure of web pages
  - Multiple pathways to internal and external pages
  - Paths for different levels of difficulty (novice, advanced, expert). For example, in Figure 1, Ross uses the analogy of ski slope runs as an indication of difficulty levels with green circle for novice, blue square for advanced and black diamond for experts.
  - Ease of access to different content areas
- Aesthetic aspects of hypermedia
  - Attractive cover page, standardized pages
  - Consistency of fonts and application of colors

Figure 1: Example of Table of Contents and Levels

The visualization systems incorporated into hypertextbooks may contain the following from the sections on teacher and student support (see Sections 4.1 and 4.2):

- Collections of well documented and ready-to-use visualizations
- Seamless integration of visualization into the text
- Increased engagement of students in the visualization and in assessment
- Integration with a course management system to enhance the attractiveness and usefulness of the VizCoSH for instructors
- Consistent and multimodal visualization user interface
- Internationalization of information

Figure 2, an image from Ross's *Snapshots of the Theory of Computing* hypertextbook [49], depicts several didactical aspects of our VizCoSH model. Note the text above the automaton editor, which documents a description of the concept introduced and gives details of the visualization. This text is included in the Figure to illustrate the combination of text and dynamic content, not to enable the reader to read it; due to the layout restrictions of the report, it may not be readable. The engagement is achieved by the construction of the automaton and the ability to enter input into the model (the *changing* level of the taxonomy). Also note the smooth integration of the visualization as the applet appears embedded in the same page as the text.

In the next section, we consider different issues of hypertextbooks that must be taken into account by a potential author of a VizCoSH. We briefly describe these issues, also giving pieces of advice based on referenced successful experiences. Figure 3 shows the relationships among our vision of a VizCoSH and

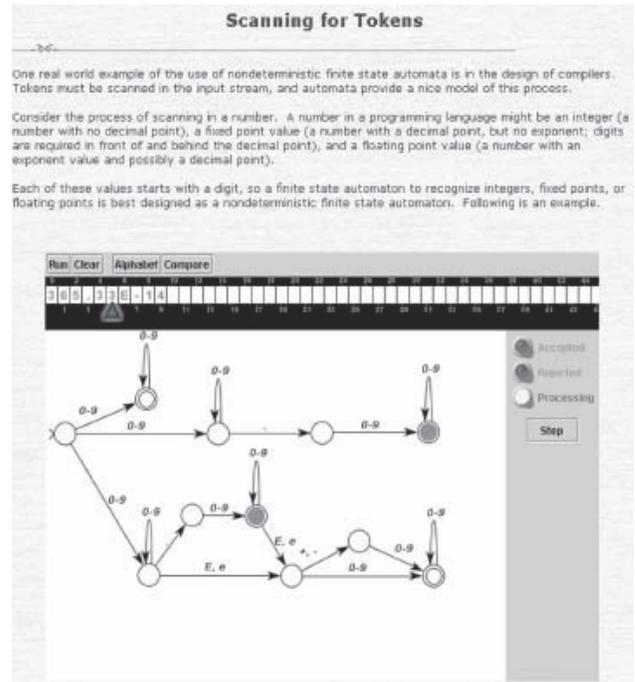adaptive hypermedia, traditional hypertextbooks and course management systems (CMS).

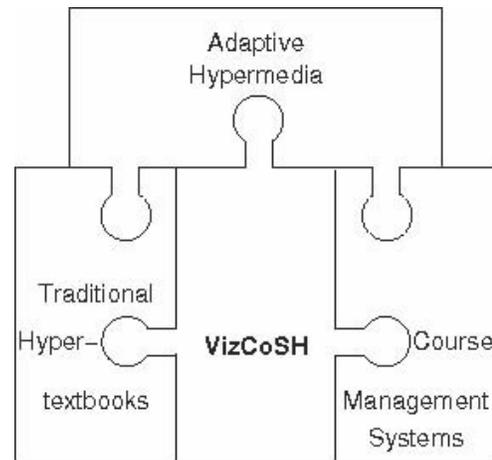**Figure 2: Example of Didactical Aspects**

Figure 3: Relationship between VisCoSH and other techniques

3.2 General Issues

The first issue to consider by a potential author is the overall goal of the hypertextbook. It can be aimed at theory, exercises, reference, etc. For instance, in the field of programming and algorithms, many hypertextbooks are theory-based while others are problem-based [38].

Based on our definition, a hypertextbook should provide, at least, the basic structure of textbooks: a table of contents and grouping of the contents according to some hierarchical structure (chapters, sections, etc.). This is especially important for novice students with no previous knowledge of the subject matter.

Instead of or in addition to written text, sounds or spoken text could also be used to accompany the visualizations, as described among other references in [22, 33, 23]. In this way, the user does not have to use a single sense (sight) to take in the content. As the visual input channel is also used by the visualizations, using sounds may reduce the user's overall cognitive load. However, this aspect could not be researched in depth by the Working Group.

The hierarchical structure typically has several levels of nodes, where the leaves are small pieces of knowledge that are atomic from the point of view of the hypertextbook author. The contents of a leaf must be displayed in a connected way. This can be achieved in a number of ways: by splitting the content into several related pages, by keeping it in the same page either in an unstructured way or splitting it into several parts accessible by a local index, and so forth. Additionally, the user's progress or "preferred" pages may also be integrated with a personal learning diary.

A hypertextbook must also consider quality and aesthetic aspects. Some good practices from the publishing community must be respected in order to achieve quality and consistency. Examples of issues to consider are design of an attractive cover, high-quality written text, standardized page design, common font usage, and consistent application of color [50]. Consistency must be extended to any other elements included in the hypertextbook, such as animations or the kind of exercises included. Good practices from the hypermedia community also must be observed, for instance, avoid using distracting elements such as flashing symbols, animated images, excessive font and color changes, etc. The length of the path to each leaf is also a concern. For instance, some authors propose to restrict some leaves to fit a screen [26, 55], whereas others provide the user some control over the amount of information displayed [55, 56].

There is a wealth of other aspects relating to aesthetics and usability that could be taken into account for an "ideal" VizCoSH. However, addressing these aspects was beyond the scope of the Working Group. Interested readers are pointed to publications in the *WWW* and *CHI* conference series, as well as to a recent book on aesthetic computing [20].

3.3   Navigation

We have mentioned the necessity of providing access by means of a table of contents. In order to facilitate reading, sequential navigation through the textbook should also be provided, at least within the same section. One of the main advantages of hypermedia is the possibility of establishing alternative structures and novel navigation models.

Hypertext can be structured on a syntactic or semantic basis. The syntactic approach admits different structures: linear, tree, graph, clustering, and so forth. Furnas [21] noted the advantages of hierarchical structures and orientation data to access information. It is probably more satisfying to complement a syntactical structure with semantics. Thus, the contents can be divided into several categories. For instance, Ross gives three different classes of contents based on the level of difficulty [50]. Hitz and Kögeler [26] distinguish between lessons, reference, and pragmatics in their book on C++, with lessons forming the skeleton of their hypertextbook while the other two categories of pages are additional material. Other common categories are exercises tied to the leaves in theory-based hypertextbooks [1, 26] or fundamentals in problem-based ones [25].

The power of hypermedia can lead to a user's becoming confused (commonly named the "lost in cyberspace" problem). One way of avoiding this problem is to provide information to users about where they are, for instance by displaying the path in the hierarchical structure followed to that particular location (often called a "breadcrumb trail") or a convention to represent the kind of information provided in that page. Direct access to capstone parts of the textbook is a way to arrive at a more comfortable location. Some common targets are "home", the table of contents, or the start of the current chapter or section. Finally, short complementary contents can be displayed in a separate window, in order to reinforce their secondary nature.

Some ways of providing richer access facilities follow, but they should be used judiciously. Direct access can be achieved by means of different kinds of indices (key terms, visualizations, exercises, etc.) as well as an internal search engine. For instance, Ross argues for providing applets that are both seamlessly integrated with text and comprehensively indexed [50]. Hypermedia also gives the flexibility to access other parts of the hypertextbook by following links in random order. Common targets of these links are chapters and sections, exercises, full views of images, special interest information, and so forth. Finally, a hypertextbook need not be a closed product, but can give access to external resources (web pages, applications, and so forth).

3.4   Adaptation

Adaptation can cover two different views: the user adapts the VizCoSH to his or her needs, or the VizCoSH tries to adapt to the user. The first adaptation step can be performed by any user on a low level, e.g. by changing color preferences. A more relevant adaptation regards providing a special "learning path" that creates the user's "personal" path through the provided learning materials. This adaptation is usually performed by a teacher or student who wants to ensure that the VizCoSH presents the same set of contents as his or her course.

The second type of adaptation concerns approaches where the VizCoSH tries to better adapt to a given user, for example by presenting certain additional information or skipping already treated information. Some of these aspects are researched in the area of adaptive hypermedia, e.g. in [52], where the learning system adapts the course navigation to the user's preferred learning style.

Both adaptation approaches require the user to be logged in when working with the VizCoSH. They usually also need a more refined user model that represents the user's preferences, and may also need to model the user's gained knowledge in different areas. Due to the restricted amount of time available to the Working Group, we can only provide a few hints on adaptation in this section.

Typically, readers adapt a book to their needs. Access to the contents of a hypertextbook may also be adapted to a particular course or teacher. We describe several ways of adapting a hypertextbook. The adaptation is performed either by using the course syllabus or even by the hypertextbook itself, drawing upon techniques from the field of adaptive hypermedia. Contents can be adapted in different ways, which can be mapped to their hierarchical structure: leaves (high level of detail), internal nodes (medium level of detail), and the root node (low level of detail).

Leaves represent atomic elements such as visualizations or textual descriptions. Both can be adapted by means of controlling the amount of information they present (summarized versions of

text, or execution stages hidden in the program animations). The control instrument should respect the student's demonstrated knowledge [8]; for instance: do not show the detailed execution of "swap operations" if the reader already understands it.

Internal nodes of the structure can be understood as groups of information. Teachers might want to use an alternative terminology, give extra explanations or add examples. This could be achieved by adding notes and self-produced pages to the hypertextbook [24]. Although previous adaptations are performed by the teacher, students can also add relevant information (for example, notes or bookmarks).

Finally, the root node represents the entire hypertextbook, so its adaptation can be understood in terms of adaptation to the goals of the course. This affects what should be explained and what should not, and also the order of study [8]. Different navigation paths through the contents can be defined. They can be based on the knowledge level of the user; but also on the level of difficulty, the goals of the course, etc.

3.5 Didactical Activities

Hypertextbooks are more than deliverers of information. As one of the main objectives is engaging students, other activities rather than just reading/viewing have to be supported by hypertextbooks. Didactical activities should be viewed at two levels of detail: course organization and learning materials.

From the point of view of course organization, exercises and virtual laboratories are good ways to achieve student engagement. Exercises make possible the responding level of the engagement taxonomy. Virtual laboratories provide the students with the opportunity to use a number of tools (simulations, animations, and so forth). Depending on their nature, exercises and virtual laboratories can be placed at different engagement levels.

From the point of view of learning materials, interaction possibilities can be augmented. Thus, exercises do not simply state a problem to be solved by the student. They also provide students with tools to solve the problems, for instance, a scientific calculator for a physics problem, or compile/execute facilities for a programming problem. These tools can be made available by integrating them into the hypertextbook itself [50] or by providing a link [1] or a gate to them [26]. Questions about the contents may have some kind of assessment incorporated, so students can be aware of their knowledge. A feedback facility can also be included. This allows students to know why they were wrong, to reinforce their knowledge by accessing additional advanced information if they were right, and help them by means of cues and other alternatives that provide more detailed explanations of the question. Animations would not be just movies played on the screen but would instead demand from the students a higher engagement by controlling the pace and direction, answering questions, changing input data, constructing their own animations, and presenting those animations to other people [44].

If students are able to upload their own content, an appropriate way to handle access and ownership has to be established, e.g. by adopting one of the guidelines offered by the *Creative Commons* organization (see http://creativecommons.org for more details).

3.6 Additional Features

Hypertextbooks should take advantage of other different communication channels beyond a mere textual channel. Multimedia provides a variety of communication channels that should be used: sound, video, and so forth [50]. The inclusion of other media should be done in a consistent way. The risk of overloading the user with too many different incoming messages on different channels must be avoided [2], although this is the author's responsibility.

4. DIDACTICAL ASPECTS

4.1 Teacher Support

In this section, we reflect on the type of support that a VizCoSH provides for teachers. These reflections address how a number of the problems cited in Section 2.2 are mitigated, how the VizCoSH promotes teachers reacting to student performance, and an example scenario that integrates many of the aspects of teacher support.

The VizCoSH combines a seamless integration of visualizations, elements for increasing student engagement, incentives for teacher adoption, and some aspects of a course management system. In the following sections, we will outline each of these aspects. An example scenario illustrates how a teacher without much free time to devote to new technology can benefit from using a VizCoSH.

4.1.1 Seamless Visualization Integration

The integration of hypertextbooks and visualizations in order to form collections of well-documented and ready-to-use visualizations is key to our approach. This approach can be effective only if the quality of the textbook elements of the VizCoSH is at least at the quality of the best traditional paper texts available in the subject area.

One of the major concerns that teachers mention for not using visualizations – namely, allocating time to search for appropriate visualizations – is eliminated because the hypertextbook provides them with a collection of visualizations relating to topics in the text. The teachers will not have to spend their time in the classroom giving explanations on how to use the animation and how to interpret what the students see on the screen. These explanations will be incorporated into the hypertextbook.

In practice, teachers can assign introductory material as a homework assignment, gaining time to focus their classroom pedagogy on more interesting topics. For instance, the classroom lecture could focus on the proofs of the correctness of an algorithm, its complexity considerations, the limitations of the algorithm, and so on.

In summary, the approach will make it easy for teachers to integrate visualizations into their teaching approach when the VizCoSH provides a sufficient number of visualizations with appropriate content. Of course, even a high quality visualization may not highlight what some teacher feels is a key step in the algorithm.

4.1.2 Increasing Student Engagement

In our concept of a VizCoSH, the visualization component plays a major role. This component provides a mechanism for increasing the engagement levels of students, allowing them to be active participants in the learning process. The visualization component addresses the concern that students are too passive by supplying questions that engage the students during the visualization itself and providing online quizzes. Later we will discuss the associated course management elements that would provide this feature as well as other potentially useful pedagogy.

In the taxonomy found in [44] and discussed briefly in Section 2.2, *responding* is the first activity that involves the student in the visualization. Queries directed at the students force them to

demonstrate an understanding of the algorithm. This feature of the visualization component is enhanced by providing feedback when the student fails to respond correctly to a question. This could be a simple hint provided in some message area (dialog box) or it could be augmented with access to additional textual material related to the question. Correct answers could also be reinforced in a similar manner.

The *changing* level expects students to have some control over the data that directs the visualization, requiring a more intimate understanding of the algorithm. To integrate the *construction* and *presentation* levels of engagement into the VizCoSH, tools exist that significantly reduce the complexity of student construction of visualizations [56]. These tools should promote the use of the construction level of engagement. If a student and/or teacher decides to develop the visualization within the visualization component, this can require considerably more effort on their part. The student or teacher would need to possess a mastery of the visualization system employed and be provided with a mechanism for adding their content to the VizCoSH. Using the facilities noted above becomes especially important when the hypertextbook does not serve as the primary text for the course and is used as a supplement. This use is a bit problematic in the sense that now the teacher's favored algorithm implementation may not be available in the visualization component and would need to be supplied to make use of it in the classroom or as an assignment.

An additional concern expressed by teachers that is addressed by our approach is the fear of spending too much time on learning a lot of different tools, especially visualization tools. To address one aspect of this issue, the VizCoSH can offer a user interface with a unified look and feel that allows the users of visualization to see the same control panel for the different visualizations.

4.1.3 Providing a Richer Learning Environment

It is natural to ask what features are implicit in the visualization component or could be added to it to enhance a teacher's desire to use a VizCoSH. These features might benefit the students by providing a richer learning environment or for the teacher by making the pedagogical experience easier and more effective. A few possibilities are discussed next.

A VizCoSH provides the possibility of incorporating new kinds of exercises. The teacher has the opportunity to assign exercises that can only be answered when the students use the accompanying visualization component. For example, the students may have to provide special input to the visualization and observe what happens in order to be able to answer corresponding questions from a quiz. It is even conceivable that the answering of questions is integrated with the visualization. The animation may stop at some point in time, and display a question. Depending on the type of question and the capabilities of the system, the question may be a multiple-choice question or a question that must be answered visually. As an example, consider the often cited example of sorting algorithm animations. The question may be to identify which elements will be exchanged next. This question may be answered visually by just clicking on them in the animation window.

4.1.4 Integration of Selected CMS Features

A central part of the VizCoSH is a course management system that interacts with the hypertextbook and the visualization component. It allows the teacher to track student use by performing such activities as monitoring the visualizations they have accessed, how often they have used them, how well they do

in responding mode (giving hints about the VizCoSH's effectiveness as a learning tool), and storing the results of online quizzes, and so forth. This data could then be used to effect adaptability.

This adaptability could provide teachers with a variety of ways to control the flow of student access, including enforcing deadlines, restricting access to links until certain goals have been achieved, and developing learning paths for students.

Another element of the course management system that could enhance the learning experience is the availability of a tool to promote communication between students and their teacher. For example, teachers could use it to provide feedback to the students, for two-way communication with students concerning problems encountered, or to simply allows students to ask questions of one another.

4.1.5 Scenario Example

To meld a number of these ideas together with the concerns they address and to see how the VizCoSH helps alleviate the concerns, we provide the following example in the form of a scenario.

Teacher X is teaching data structures and algorithms. Teacher X believes that visualizations should help students, but has a heavy teaching load and worries about the time needed to:

- Search for visualizations

- Explain to students how to use the visualization system

- Adapt visualizations to teaching approach and/or course content

- Transition the visualizations into viable exercises for students

- Install the software

A VizCoSH on data structures and algorithms provides links to visualizations produced in real-time, all of which use the same visualization system. Several pages in the VizCoSH are dedicated to providing an overview of the perspective the VizCoSH takes on algorithms and data structures in general. That is, are the chapters in the hypertextbook organized by type of data structure (stack, queues, trees,…) or by algorithmic technique (dynamic programming, divide-and-conquer, greedy algorithms,…)? There are pages in the hypertextbook that explain how a user of the visualization system should navigate using the GUI of the system. The treatment of each particular algorithm in the hypertextbook is organized along the following lines:

- Learning objectives for the student in this section

- A conceptual discussion of the algorithm being studied

- Advice to the students about how the preceding conceptual discussion may differ in minor details from the treatment given in their regular textbook – if they were required to buy such a textbook in addition to the VizCoSH.

- Hyperlinked text that will launch the visualization system's animation of the algorithm. Here the student can launch repeated instances of the animation, each acting on a different data set. Each instance is accompanied by text explaining the significance of the graphic artifacts the student is viewing on the screen during the animation and a set of questions the student should answer during the course of the animation.

- Exercises that the student completes after viewing an appropriate number of "practice" animations. These exercises test how well the student learned what the animation was intended to show.

Note how the features and organization of the VizCoSH specifically address the time-related worries of the teacher. The teacher no longer has to search for visualizations since they are all bound together by the hypertextual glue provided by the VizCoSH. The visualizations are launched from within the hypertextbook (applet, WebStart, Flash,…), relieving the teacher from the burden of installing the visualization software. The text in the VizCoSH takes care of "setting the stage" for the student by providing explanations of algorithms and how the graphic artifacts of the visualization reveal information about the algorithm's execution. Students have questions to answer as the visualization progresses and follow-up exercises to complete and test their understanding after the appropriate number of visualizations have been viewed.

4.2 Student Support

This section discusses the possible problems and requirements of students using a VizCoSH and how the VizCoSH can address these issues. From a student's perspective, a good VizCoSH consists of two design dimensions of the provided information – the text part and the visualization/animation part. These two dimensions are strongly affiliated with each other. First, we discuss the text part in more detail.

4.2.1 Textual Components

Most people prefer a kind of *atomic approach* in the text design of a hypertextbook, which means that each knowledge entity should be described and explained on a single page. An example for this is the GaniFA hypertextbook [15]. Here, each step of the generation process of a finite automaton from a regular expression is treated separately. Thus, the student has all relevant information available to understand the actual learning topic. Of course, this could contradict the basic hypertext idea because of the possible duplication of issues that are discussed elsewhere in the textbook.

In this context, students expect that a VizCoSH is *written consistently* in many respects:

1. The writing styles of different authors should be unified.

2. The abstraction levels of all text parts should be standardized. This is important for the definition of difficulty levels that can be adapted by the student.

3. Use of definitions, abbreviations, and mathematical symbols must be consistent in the entire VizCoSH. Changes to these are not likely to be acceptable to the learners. This is more a matter of consistent notation as visible to the student rather than its encoding in the hypertext pages. For example, the encoding could be done in MathML [10], which offers great and far-reaching support. However, only Mozilla-based web browsers currently support MathML natively.

4.2.2 Visualization Components

The design of the visualization part itself is one of the most important issues of VizCoSH: Visualizations and animations should be *smoothly integrated* into the text. Opening new windows in order to display visualizations would break the flow of understanding the subject. Here, we can learn from the well-known *Focus-and-Context* techniques in information

visualization: Visualizations and animations should be displayed within their textual context without overlapping windows. This also guarantees that visualizations appear close to their describing text.

The appearance of all visualizations/animations should also be unified (*Common-Look-and-Feel strategy*). This is important for the control elements (for example, the visual representation of the VCR metaphor to control an animation – see JHAVÉ [41] and Jeliot [39] for more information) but also for the animations themselves. Semantically identical elements of an algorithm, for example variables, should have an identical visual representation. Additionally, this is important for different interaction techniques offered by mouse or by keyboard. For example, a right mouse-click should force the same behavior in all interactive visualizations.

4.2.3 Consistency and Aesthetics

Students need to see the *same terms* in the text *and* in the visualizations. A simple embedding process of different visualization resources does not suffice in most cases. If this aspect is not guaranteed, many students may give up quickly because the translation takes too long.

In the visualization community, *aesthetic aspects* are becoming important [20]. To underscore the importance of aesthetics, the first Eurographics Workshop on Computational Aesthetics in Graphics, Visualization and Imaging has taken place in Girona, Spain, 2005. Aesthetic aspects are also essential from a student's perspective. The visual appearance of the text (font, use of colors, design of tables and diagrams, and so forth) should be clear and modern. The same is true for interactive visualizations and animations.

4.2.4 Motivating Students to Use the VizCoSH

A further key issue is to motivate students to use the VizCoSH. Ideally, the combination of good text materials merged with visualizations and interactive exercises will be sufficient to keep the students' intrinsic motivation for using the VizCoSH high. However, we have to consider that some students' intrinsic motivation will not be sufficient, and extrinsic motivation will be required.

Probably the best way to motivate less keen students to use the VizCoSH is to use it in the *grading* of the course. The grading can be based on the student's usage of the visualizations. However, it is obvious that only page visits or passive viewing of videos cannot be used for evaluation. Thus, material on at least the responding level of the engagement taxonomy is required. With the visualization focus of the working group, the most obvious form of this graded material derives from visualizations. However, it can be any assessed part of the material.

The use of the VizCoSH as compulsory material in a course to motivate the students is not always feasible. Thus, the material should also motivate student use purely because it is interesting and aids their learning. The use of interactive visualizations offers a way to better engage students with the material than a traditional textbook does. It can be argued that this interaction can make learning more interesting, or even fun, for the students. However, this requires the visualizations to be on a higher level than viewing in the engagement taxonomy.

To avoid becoming lost in the material, the students should always see the overall context in the structure of the book where they currently are. For the students to be aware of what they have learned (or should have learned) the VizCoSH should support a *visible learning process*. This could, for example, show students

what topics they have already studied and how well they have performed in the interactive mode.

Furthermore, the system can provide students with recommendations of required prerequisites of a topic or suggest suitable topics after learning one. In addition, traditional textbooks make it easy to add notes next to the text to help the understanding of a topic the next time the student reads the text. In a VizCoSH, this can be achieved by allowing *personal annotations* by students, similar to the functionality offered by Tablet PCs.

4.2.5 Use Across Cultures

Because computer science education crosses many cultural and language boundaries, it is important for the students to have learning material in a language they can understand. Preferably, this would be their native language. This requires *internationalization* of the material in the VizCoSH, so that it can be offered in a variety of different languages.

Apart from translating contents, the materials may also have to be put into the proper cultural context (*contextualization*). For example, different cultures may have different views on reading direction (for example, compare English with Arabic or Hebrew and some Asian languages) or the meaning of colors (red is associated with "fire, danger, warning" in some cultures, but with "warmth" and other positive connotations in others). Restricting the translation to the words, and where necessary, also the character encoding (e.g., Chinese characters) or reading direction, may therefore still confuse some students.

4.2.6 Adaptability

Students from different cultures or students who have a significantly different *educational background* share similar problems. The VizCoSH must account for different levels of preparation and expectation. In this case, the students can have very different previous knowledge of the topic as well as different needs in learning the material.

Furthermore, the students can have different learning styles [19, 18]. Different students can be better served by a VizCoSH if the material supports *adaptability*. This adaptation can be based on various factors. For example, it can be based on the previously read pages, on the interaction with the visualizations, or on the difficulty level selected by the student/teacher. The effects of the adaptation can be anything from a small change in the level of detail of a visualization to a completely different animation.

Paradoxically, the combination of adaptation and all the other advanced features of the VizCoSH can hinder the usability of the material. To prevent this, it must be made certain that the features aid the students instead of confusing them. In other words, *usability* of the end product must always be kept in mind when adding new features.

4.2.7 Multimodality

Supporting several input channels, that is, *multimodality*, can play a decisive role. In cognitive psychology, research has shown that the support of visualization by sound or auditive explanations can improve the understanding of the learning subject [57, 29]. Furthermore, it could also enhance the learning of students with other learning style preferences. The VizCoSH could provide access to audio that supplements the visual components for the visually impaired.

4.2.8 Constructivism

Novel educational systems follow the mantra of *constructivism* – see for example [30, 34, 16]. The constructivist approach is based on the idea that the knowledge has to generate itself in the learner's mind. Therefore, knowledge cannot be transferred in a traditional lecture or reading mode. Within moderate constructivism, the teacher, the expert and the system are not allowed to manipulate the learner's construction process but they can offer help and coach their individual construction processes. Therefore, a goal of moderate constructivism is to build learning environments that give learners the possibility to generate their own knowledge constructs.

One possibility to reach this goal in the context of learning algorithms is to use compiler generation techniques to generate interactive algorithm animations from specifications [31, 32]. A qualitative evaluation based on this work also shows that students want to explore with visualizations, try out different ideas, check hypotheses, and so forth. Thus, constructivist ideas and visualization tools/applets realizing such concepts are very interesting for the learner in a manner that is much more subjective than traditional teaching methods in computer science.

5. TECHNICAL ASPECTS

The previous sections have introduced a long list of features that would make a VizCoSH an attractive option for teachers to use as part of teaching as well as an aid to motivate students in their learning. In this section, we will discuss the different technical aspects related to the actual implementation of a VizCoSH.

5.1 Visualization Integration

The crucial technical aspect is naturally the integration of visual content into the VizCoSH. There are several different options to do this. In the following, we will mention some options and discuss the drawbacks and benefits of the different approaches. Possible options include but are not limited to:

Video material can be very easily embedded. However, the creation of video material is very time consuming (unless it is generated by recording a visualization system). Moreover, to show video material the user needs the relevant application or browser plug-in. Such plug-ins (e.g., QuickTime) are ubiquitous, and it is likely that students will already have the most common video plug-ins installed.

Flash content offers a technique that is used on a multitude of web pages. One benefit compared to the other solutions is the relatively small size of Flash animations. Drawbacks of Flash animations are the effort needed to produce animations, the required browser plug-in, and the commercial product required to author the animations.

Java applets are probably the most often used technique to embed algorithm visualizations in online material. A vast number of existing systems [51, 37, 41, 47] use applets. There are drawbacks with applets as well. First of all, the security restrictions make the usage of local files difficult. Furthermore, as applets embedded into a page are loaded once the page is loaded, the loading and startup times of a single page can be too long. In addition, despite the platform independence of Java, there have been problems with applets working differently on different platforms.

Luckily, these problems are less prevalent than they once were.

Java WebStart allows the embedding of fully-fledged Java applications. If properly done and the user trusts the author, the WebStart application can do everything a normal locally executed Java application can.

Since a primary aim of a VizCoSH is to engage the student in learning, considering the engagement levels supported by the different techniques is important. Video material supports only the viewing level of the engagement taxonomy. The higher levels of the taxonomy can be achieved with the three other options. How-ever, currently existing solutions supporting the higher levels of engagement are mostly Java-based [51, 37, 39, 41, 47].

To use the VizCoSH in course grading, the technique used to embed visualizations must support assessment. This is possible to achieve in all approaches besides video material. However, as with engagement, existing solutions either use Java applets or Java WebStart. Examples of such systems are JHAVÉ [41] and lately also Jeliot [40] where the students are asked questions during a visualization. The answers to these questions can be automatically graded. Another tool is Trakla2 [37] where the students are usually given a set of data and asked to construct an animation by following a given algorithm. This can be, for example, inserting a set of keys into a binary search tree. The animation constructed by the student can again be automatically assessed.

To use or show the grades in the VizCoSH, the visualizations have to communicate with the VizCoSH (or the related course management system). To achieve this, there has to be a common communication protocol to send messages between the parts. Such a protocol can be a simple protocol based on HTTP and XML, as the protocols proposed by Brusilovsky et. al. [9]. A similar approach is used by JHAVÉ [41]. Another example is Trakla2 [37] that uses XMLRPC for the communication. This kind of communication could be implemented in Java and Flash-based solutions.

5.2 Adaptation

For the VizCoSH to support adaptation, one imperative feature is to have some sort of student model. This model then has to be updated whenever a student interacts with the VizCoSH material. Student modeling has been an active research topic, for example, in adaptive hypermedia. Therefore, visualization system designers may avoid implementing their own student modeling facilities by perhaps adopting one of the existing solutions. One such solution is CUMULATE [9] which is a student model server that supports updating and querying the student model.

In addition, the material must be categorized and annotated for the adaptation to be possible. For example, for the VizCoSH to be able to adapt to the knowledge level of the student, the environment must have knowledge about the difficulties of different parts of the material. For this, there is a need for a common ontology for the different concepts in the VizCoSH.

5.3 Distributing the VizCoSH

There are several options available to distribute the VizCoSH. The easiest way for teachers only wanting to use the VizCoSH as additional material supporting the students is to have the VizCoSH available on one website. However, to use the VizCoSH in grading or modifying the material, the teachers will most probably want to have a local copy of the material. This local copy can be obtained in different ways. It can be, for example, downloaded from a website or checked out from a CVS/Subversion repository. Downloading is perhaps the easiest solution for a teacher. However, a version control system might be a more sensible solution if many teachers want to and are allowed to share their modifications with other users of the VizCoSH. This depends on how the editing of the VizCoSH "master copy" is organized. For truly responsive VizCoSHs that incorporate large images, video clips, and other download intensive components, it would be best for students to have local copies on their hard drives.

5.4 System Design and Further Aspects

The actual architecture of a VizCoSH that implements the features described in this report as yet remains to be defined. The depth of research and software design needed for this was simply beyond the scope of the 2006 ITiCSE Working Group. For example, the reader will have noticed that our work was largely inspired by the research done by Ross and his group [5, 4, 13, 49, 48]. However, other approaches, especially those based on established standards such as *Learning Object Metadata (LOM)* [36] and *SCORM* [53], need to be taken into account.

The design for a VizCoSH can be determined in follow-up research projects both within and without the Working Group. Similarly, more detailed aspects of a VizCoSH, such as community building and maintenance issues, as well as the challenges facing open courseware, could only be addressed briefly by the Working Group. Our first answers to these topics can be found in Section 7.

6. SCENARIOS AND USE

6.1 Teachers Who Want to Contribute

Some teachers would be willing to devote more time to other activities if the VizCoSH supports them. Some examples follow, most of them requiring the VizCoSH to include facilities or complementary tools to modify the VizCoSH contents.

If the VizCoSH can be customized, teachers might wish to customize its contents to their needs or preferences. This could be achieved by just writing a user manual with an alternative path to follow in their course, similar to that found in the preface of many textbooks. Typographical features of the VizCoSH constitute another opportunity for customization: headings, font, page back-ground, and so forth.

If the VizCoSH permits it, more demanding changes can be made by an instructor on the order of the VizCoSH topics, even hiding the irrelevant parts for their course. Changing the contents of the VizCoSH will probably be a rarer feature. However, an annotation facility may allow a teacher to include comments on selected parts of the VizCoSH. These comments may deal with terminology, elaborating or inserting a new part of a section, and so forth.

For inserting new sections, an editing tool must be provided so that contents can be inserted in the VizCoSH and the table of contents can be modified. Notice that most of these facilities require using a technology that allows keeping contents and presentation separate (for example, XML). Moreover, the teacher will have to devote the time necessary to produce the contents plus the time to use the publishing tools.

The visualization part of the VizCoSH also provides opportunities to improve it. Note that teachers usually have different preferences on how to explain given topics.

Demonstrations inserted as examples in given parts of the VizCoSH could be modified to consider a particular example by providing specific input data. As cited above, a more advanced facility generates text as well as animations of new "entities" (for example, finite automata or algorithms). A complementary feature, mentioned below, is the possibility of creating and collecting student-generated animations.

6.2   Novice Students

Novice students will have little or no experience with visualization tools, whatever engagement levels are considered. In this scenario, we discuss the viewing, responding, and changing engagement levels for such students.

First, students will be able to access visualizations by just viewing them, either during classes or in their home. It is supposed that they will become quickly familiarized with the typical VCR-like interface of these animations. So they can "use" both static textual contents and dynamic visualization contents in the VizCoSH. Then, they can be asked during the animation, at the responding engagement level, to answer questions about concepts displayed or to predict algorithm behavior.

Now, students can assume a significant change in the way they interact with visualizations. Students will be able to change what is displayed in the visualization by changing its input data. Actually, students change the visualization in an indirect way, because they change the behavior of the algorithm.

Notice that novice students are working directly with the algorithm itself – viewing how it works, answering questions about it, or changing its input data. They are not concerned with technical issues related to the production of the visualizations. Thus, they will use (hyper)textual contents and visualizations of the VizCoSH.

Once they are familiar with visualizations and these three engagement levels, they become advanced students, and can proceed to the constructing and presenting engagement levels.

6.3   Advanced Students

Based on the *engagement taxonomy*, this scenario is for teachers who feel that students will learn more about an algorithm or data structure if they have to construct or present a visualization of it. This scenario certainly only makes sense for students familiar with the VizCoSH and, in particular, its visualization system.

Creating a visualization at first does not appear to be at the core of a VizCoSH, as this is mostly concerned with "prepared content." Still, the VizCoSH can effectively help students with this task. The VizCoSH will contain the relevant information needed to inform the students about the underlying algorithm or data structure. Additionally, a segment of the VizCoSH will be dedicated to information on how to use the different visualization tools, and how to generate new content for them. This should enable students to work on such assignments.

Once a student figures that he or she knows how to generate content, the associated visualization system can be launched from the VizCoSH using, for example, Java WebStart. The student will continue to access the VizCoSH while he or she is generating the new content. This access to the VizCoSH may be to look up details, check on the hints for generating content, and to compare the example output provided in the VizCoSH with the output of the visualization content the student has generated so far. The construction of the actual content can also be done inside the VizCoSH, if the visualization system is properly integrated into the VizCoSH.

The VizCoSH can also assist in the presentation activity by allowing the delivery of a student report including both a textual explanation and a supplemental animation.

Once the content is generated, the VizCoSH should provide a way to view the content inside the visualization engine(s) embedded in the VizCoSH. At this point in time, this acts as a "preview" of what the content could ultimately look like if it were incorporated as new dynamic content for the VizCoSH. Additionally, the student should be able to upload his or her visualization file – in the tool-specific format – and provide access to VizCoSH users. This can be in the form of a general repository, where users can find all content for a given topic or topic area.

Additionally, there should be some way for a teacher or administrator to allow the inclusion of the link to a concrete visualization on a given content page or, more probably, in a given index. Here, it is more useful to provide facilities to create and manage whole collections of animations, not only individual ones [56]. Note that a student who tries to work on an exercise will essentially follow the same process as a teacher who wishes to provide a new visualization. The difference in roles between teacher and student here essentially is that the teacher will usually have the authority to add content to the "official" version of the page, while a student may be restricted to adding a link to "related, alternative content."

6.4   Peer Reviewing Visualization Content

Use of a VizCoSH is especially suited for a distant-learning environment. If teachers want to go beyond the construction level (see Section 6.3), it may not be obvious how the presentation level can be reached in such an environment. In this case, the student presentations may be accomplished by peer reviews [45, 54].

Imagine the following scenario: Students study a certain topic that is described in the VizCoSH. Among other assignments, they have to construct their own animation, which mirrors a certain mechanism that they just studied. One or more possibilities for constructing the animation are also described in the VizCoSH.

As soon as the students have constructed their animation, they upload it to a server (phase 1 of the review process). The interface for uploading the animation is supposed to be a part of the VizCoSH. Other students then have to review one or more of the submitted animations. The reviews again have to be submitted through a VizCoSH interface (phase 2 of the review process). Finally, the students are asked to look at the reviews of their own animations, and in addition at other reviews for those animations that they had reviewed. Depending on how much time the teacher wants the student to spend for this assignment, the students may be asked to adjust their review based on the other reviews (thus returning to phase 2 of the review process), or to modify their animations based on the received reviews (thus returning to phase 1 of the review process).

Depending on the number of distant-learning students and the preferences of the teacher, there are a number of variations for the review process:

- Only a subset of the students constructs an animation for a certain topic, or all students construct an animation.

- A constructed animation is reviewed only by a subset of the students, or by all other students.

- When reviewing an animation, the reviewers know who constructed the animation, or the author of the animation is anonymous for the reviewers.

- The authors of the animation know who wrote a review for their animation, or the reviewer is anonymous for the authors of the animation.

Even if there are doubts that higher levels in the engagement taxonomy lead to significantly better learning results [35], we claim that there are positive effects of this peer review approach that go beyond mere learning issues. Especially, we claim that the following soft skills will be improved in this peer review scenario:

- Communication and cooperation between distant-learning students will be enforced and improved.

- Students will learn to constructively criticize, especially by comparing their review with the others reviews, and they learn to accept constructive criticism.

6.5 Mixed Scenario

In this scenario, we combine two of the previous scenarios to cover all the levels of the *engagement taxonomy* using a VizCoSH in a mixed setting of lectures, homework, and laboratory work. Teachers who lack time use the VizCoSH to support their lectures. They obtain the slides from the VizCoSH and present the contents of the slides in the lecture. The slides will contain the same animations provided by the VizCoSH or pointers to them, so the lecturer can use the visualizations conveniently in the middle of the lecture and reach the *viewing* engagement level.

At home, students interact with the visualizations of the VizCoSH, which should be available to them. They explore the visualizations and *respond* to the questions the hypertextbook asks about them. As well, special tasks can be set up for the students, such as asking them to report what happens when certain unusual data is processed, forcing them to *change* the visualization, and reflect on the newly produced visualization. In this way, interacting with the VizCoSH will be potentially less "boring" than with a standard textbook.

Finally, laboratory sessions will have students *constructing* their own animations for a new concept using the visualization tools that are part of the VizCoSH distribution. Finally, the laboratory environment is the right place to have students *presenting* their visualizations.

At this point, students will have developed a deep understanding of the concepts introduced by the VizCoSH. Moreover, they will have checked their understanding either by the responding phase or the presenting one.

6.6 Programming Assignment to Construct Visualization

A teacher is teaching the "dreaded" Bubble Sort algorithm. The teacher would like to have students actually code the algorithm. The teacher's dilemma here is two-fold. First, how do we make a lecture segment on Bubble Sort interesting? Second, in programming assignments, we would often like to have students code "classic" algorithms, but assigning such will just result in their searching the Internet for existing implementations and pasting the code they find into their assignment. How do we make students really master the mental transition that must occur between the "idea" behind an algorithm and developing the code that implements it?

The solution here is to use the VizCoSH, with its visualizations of the algorithm, to ultimately lead the student to constructing a new visualization as part of their programming assignment. The teacher need not lecture about understanding the Bubble Sort at all. Instead, students are directed to the VizCoSH module on the Bubble Sort algorithm. There they begin by reading a standard textbook description of the algorithm. Only high-level pseudocode is shown as part of this description. Hyperlinked text then launches the visualization system's animation of the algorithm. Here the student can launch repeated instances of the animation, each acting on a different data set. Each instance is accompanied by a set of questions that offer the student several different versions of actual code segments and ask them which code segment corresponds to the animation they are viewing. For example, if the animation shows the Bubble Sort with the element that is being located at its correct position sinking downward in the array instead of rising to the top, the student would be forced to read the code options and decide which code fragment controlled its loop in the way the animation depicts.

Once students have demonstrated a mastery of the animated version of the Bubble Sort algorithm, the VizCoSH directs them to a programming assignment that has them develop Java code for the Shaker Sort (a variation on the Bubble Sort that has the element being located moving in opposite directions on alternate passes through the array). Additionally, students are asked to use their implementation of Shaker Sort to produce a visualization of it. To achieve this, they must use a special array class such as the *GAIGSarray* class described in [42]. Such a class, in addition to providing the usual array operations, is equipped with special methods that enable students to annotate the algorithm at interesting events with method calls that result in graphical depictions of the state of the array. Although this may seem complicated for students at this level, it actually amounts to nothing more than:

1. Understanding the protocols for the array-depicting methods (described in the VizCoSH and available in Javadoc form at http://jhave.org/developer/doc) and,

2. Then adding method calls at points where they would typically add tracer output for debugging purposes.

Both of these skills are essential for students who learn programming at the introductory level. Hence, the programming assignment not only has students engage in visualization at the *constructing* level of the engagement taxonomy, but also reinforces other fundamental skills we want them to develop.

By visualizing the state of the data structure at any (student-chosen) point, students are likely to find it easier to locate bugs in their code. Additionally, having a graphical display on the screen is usually more motivating than tracing the usual debug statements, and also gives more of a sense of accomplishment. This also makes the assignment more interesting than the more "traditional" forms, where the most exciting expected result (for a sorting algorithm exercise) is that the output displayed as a text on the console is actually sorted.

7. INCREASING INTEREST IN ADOPTION

In prior sections of the paper, we have addressed numerous aspects of VizCoSHs that have the potential, based on past evidence, of increasing the effectiveness of visualization-based

learning by students while at the same time making them a more convenient resource for teachers to use. However, most arguments have been based on largely academic criteria. There have been many educational projects that, despite the sound arguments that can be made for their use, never find their way to widespread adoption. The reasons for preventing widespread adoption are often practical. Consider, for example, some of the "reasons for not adopting visualization" that the members of the working group considered before convening in Bologna – where the main concern was about *time investment* on the teacher's part, not concern about the quality of the actual content [44].

In the following, we will address a set of questions and reservations that a potential VizCoSH user might express. Where possible within the work done by this Working Group, we also outline answers or approaches to these questions.

- *Why should I change from an established textbook to a VizCoSH?*

With a good GUI and the provision of options that allow students to print portions in a logical and cohesive fashion, a well-designed VizCoSH, as envisioned in this report, can be at least as good as a "regular" textbook. In the simplest case, the VizCoSH could simply contain the full text and images or diagrams of the textbook. This content would then be enriched by dynamic visualizations of selected content. Compared to the textbook or even a hypertextbook rendition, the VizCoSH would therefore allow students to explore more aspects of the content in a practical, "hands-on" way. Additionally, engaging elements as described previously can be added easily to a VizCoSH, in comparison to the static media of a textbook.

- *How can I determine if the VizCoSH is "at least as good" as my current textbook?*

Essentially, in the same way teachers are currently determining their choice for a textbook. This is based on a set of decisions, such as topics covered, writing style, incorporation of examples and possible exercises, availability of prepared slides or model solutions, and many others. We are confident that a well-designed VizCoSH can compare favorably with a regular textbook in these areas, and may provide additional features, such as "live" exercises with the potential of teacher feedback for grading purposes.

- *Who guarantees the quality of the materials in the VizCoSH?*

This partially depends on how the VizCoSH is distributed. If the content is edited in a Wiki-like approach, experience from large projects such as Wikipedia indicates that the content may be of a very high quality, but not necessarily stable.

If the VizCoSH is distributed by a more "official" channel, for example, access controlled by a publisher or on a fixed medium such as a DVD, there may be at least a "fall-back" version of the contents that will survive editing.

- *Will the VizCoSH still be available next term (next year, the next time I teach the same course…)?*

Again, this question ties in to an extent with the distribution of the VizCoSH. For a web-based distribution supported by the community of developing authors, it is highly unlikely that the VizCoSH will simply disappear—although its URL may change, as servers are renamed or faculty change their university affiliation. If the distribution uses a static medium, such as a DVD, its availability will be similar to that of other books, CDs, or DVDs, which may run out of print or stock. However, in this case, the teacher who used the resource the last time should still have a copy of its medium (CD or DVD).

- *Will the content remain stable and reliable?*

As addressed in a previous question, this largely depends on how the VizCoSH is set up in practice. Even if the VizCoSH is set up to be editable by a set of authors, we recommend a configuration similar to a Subversion or CVS repository that will store a copy of the previous edits. A given teacher should then be able to store their "personal view" of the contents, specifying that a certain version of a page should belong to the personal "learning path," which also becomes the path used by that teacher's students.

- *Who will maintain the VizCoSH? How will it survive if there is no fixed maintenance plan?*

We expect that the community of authors will start with the maintenance work for the VizCoSH. After a number of teachers have started adopting a given VizCoSH for their personal course use, their interest in keeping this book running will help in finding partners. Again, this touches on the question of distribution – for a freely-available resource, it may be possible to find volunteers (or student helpers); for a commercial resource, such as one provided by publishers, the interest in maintenance will grow with the number of users. To be fair, this still means that an initial ramp-up phase of acquiring a sufficient number of users may be necessary.

It is not even necessary that a VizCoSH be distributed freely. Even a modest charge (e.g., $10 instead of the more likely $110 for a traditional textbook) could help support continued maintenance of the VizCoSH.

- *What will happen if the content creator eventually leaves the authoring process?*

There will probably not be a single content creator. In similar projects where mainly one person develops content, the long time frame involved greatly limits distribution [17]. The way of distribution will probably make it necessary to place at least parts of the content under an appropriate license, such as the GNU Lesser Public License or one of the guidelines provided by the Creative Commons organization (see http://creativecommons.org). The same applies to the underlying code that ties together the VizCoSH—this also has to be placed under a license that allows users to keep accessing the VizCoSH even if the original programmer is no longer part of the VizCoSH.

- *What about special bonus content for teachers?*

Publishers of traditional textbooks offer additional bonus content for teachers, especially PowerPoint slides, prepared exercises, and example solutions. This can also be done in a VizCoSH, for example, by regulating the accessible contents according to the privileges granted to the current (authenticated) user. The presence of facilities for the higher levels of the engagement taxonomy ideally prepares a VizCoSH for generating at least visualization content that can be used directly in class, as well as for students in self-study periods.

Additional ("fixed") content, such as a set of presentation slides, or exercises with example solutions, can easily be incorporated into a VizCoSH. Depending on the output format, it is even possible to generate such content on the fly, based on the teacher's preferences or concrete data input. This would have to be based on open and well-documented formats, such as PDF or the *Open Document* format used by *OpenOffice*, which was recently officially mandated by the State of Massachusetts. The

same would apply to Microsoft's upcoming XML-based format for their office products, if this is sufficiently documented and "simple" to enable automatic content generation.

- *Are VizCoSHs a threat to the "freedom of teaching"?*

In Germany (and perhaps in the rest of Europe) teachers are more reluctant to adopt textbooks than in the U.S. The "freedom of teaching" means that teachers have the right to define the contents of their courses. Although many teachers use the same set of textbooks as resources, each teacher emphasizes different issues, and many teachers produce their own teaching materials rather than using established materials from publishers or colleagues. The "freedom of teaching" is sometimes (mis-) interpreted by teachers as a duty to produce a unique teaching unit. This attitude is an obstacle against the adoption of VizCoSHs.

A strategy to address this attitude is the creation of a critical mass that starts using a VizCoSH for a certain topic. Once this occurs, there may even be pressure coming to bear from the students who are required to use this VizCoSH with its eye-catching animations that are used in other universities. In that respect, it is important that it remain possible for teachers to add their own specific version of a given topic, so that their "freedom of teaching" is not restricted in any way. In that respect, the usage of the VizCoSH by the teachers should be different from Wikipedia, where there is only a single current definition for a term.

- *Is there any installation involved?*

Again, the answer depends on the way the VizCoSH is provided to students and teachers. For a "static" distribution, such as a CD or DVD, installation can be as easy as for other software distributed on these media. If a (single) web server is used for all users, an individual user needs only to access the appropriate URL. For running a local copy of the VizCoSH, users have to set up a web server with standard components. Typically, this will be a very well-documented installation, including Java, a web server (for example, the freely available and well-documented Apache), and, depending on the implementation, a standard freely available database such as MySQL. The dynamic aspects of the VizCoSH may involve appropriate environments, such as PHP or a Java servlet—again, these components are very easy to install, and may even be pre-installed on Unix-based web servers.

- *Once installed, how difficult is it to use?*

A VizCoSH consists of a set of hypertext pages with added dynamic content, especially visualizations. Therefore, using it will be as easy as using any other web page that combines interactive elements (such as HTML forms) and dynamic content (such as movies).

8. CONCLUSIONS

In this paper, we have formulated a vision of how various aspects of instructional visualization in computer science could be enhanced by using VizCoSHs as delivery systems. We have argued that, from a student's perspective, VizCoSHs coordinate activities whereby students are actively engaged with the visualization instead of merely viewing it. Additionally, course management components of VizCoSHs, such as recording how students do in these engagement activities, increase student motivation and allow adaptation of the visualizations based on student comprehension of the material. From the teacher's perspective, we have suggested how VizCoSHs could alleviate many of the concerns that have been previously expressed regarding the investment in time required to effectively use visualizations in their teaching. Much of what teachers would have to do to integrate visualizations into their courses could now be off-loaded to the VizCoSH. Scenarios were discussed to illustrate how the use of VizCoSHs would actually occur in a variety of instructional settings.

Much work remains to be done in realizing the goal of building successful VizCoSHs. In Section 5, we described many of the technical questions that need to be resolved. In Section 7, we addressed practical questions that teachers are likely to have about the viability of what we have proposed. Providing more robust answers to both the technical and these practical questions is critical to the eventual widespread adoption of VizCoSHs. As a means of encouraging members of this working group, other teachers, and developers of visualization systems to quickly follow up on the activities we have proposed, a website has been setup at http://www.algoanim.net/vizcosh/.

9. REFERENCES

[1]   I. Aedo, P. Díaz, C. Fernández, G. M. Martín, and A. Berlanga. Assessing the utility of an interactive electronic book for learning the Pascal programming language. *IEEE Transactions on Education*, 43(3):403–413, Aug. 2000.

[2]   M. Bearne, S. Jones, and J. Sapsford-Francis. Towards usability guidelines for multimedia systems. In *MULTIMEDIA'94: Proceedings of the second ACM international conference on Multimedia*, pages 105–110, New York, NY, USA, 1994. ACM Press.

[3]   B. S. Bloom and D. R. Krathwohl. *Taxonomy of Educational Objectives; the Classification of Educational Goals, Handbook I: Cognitive Domain*. Addison-Wesley, 1956.

[4]   C. Boroni, F. Goosey, M. Grinder, and R. Ross. Active Learning Hypertextbooks for the Web. *Journal of Visual Languages and Computing*, 13(2):341–354, 2002.

[5]   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 32nd ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE 2001), Charlotte, North Carolina*, pages 65–69. ACM Press, New York, 2001.

[6]   J. H. Brewer. The science1hypertextbook web site. Online at http://musr.physics.ubc.ca/˜htb, 2001.

[7]   J. H. Brewer. What Is a Hypertextbook Anyway? Online at http://musr.physics.ubc.ca/˜jess/HTB3/, 2001.

[8]   P. Brusilovsky. Adaptive navigation support: From adaptive hypermedia to the adaptive web and beyond. *Psychology*, 2(1):7–23, 2004.

[9]   P. Brusilovsky. Knowledgetree: a distributed architecture for adaptive e-learning. In *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 104–113, New York, NY, USA, 2004. ACM Press.

[10] D. Carlisle, P. Ion, R. Miner, and N. Poppelier. Mathematical Markup Language (MathML) version 2.0 (Second Edition), Oct. 2003. http://www.w3.org/TR/MathML2/.

[11] P. Crescenzi, C. Demetrescu, I. Finocchi, and R. Petreschi. Reversible Execution and Visualization of Programs with LEONARDO. *Journal of Visual Languages and Computing*, 11(2):125–150, Apr. 2000.

[12] F. Crestani, M. Landoni, and M. Melucci. Appearance and functionality of electronic books. *International Journal of Digital Libraries*, 6(2):192–209, Apr. 2006.

[13] A. B. Cunningham and R. J. Ross. Biofilms: The Hypertextbook. Online at http://www.erc.montana.edu/biofilmbook/, 2006.

[14] S. Diehl and A. Kerren. Levels of Exploration. In *Proceedings of the 32nd ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE 2001), Charlotte, North Carolina*, pages 60–64. ACM Press, New York, Feb. 2001.

[15] S. Diehl, A. Kerren, and T. Weller. Visual exploration of generation algorithms for finite automata. In *Implementation and Application of Automata*, volume 2088 of *Lecture Notes on Computer Science, LNCS*, pages 327–328. Springer, 2001.

[16] T. M. Duffy, J. Lowyck, and D. H. Jonassen. *Designing Environments for Constructive Learning*, volume 105 of *NATO ASI Series F, Computer and System Sciences*. Springer, Berlin, Heidelberg, New York, London, 1993.

[17] G. Elert. About The Physics Hypertextbook. Online at http://hypertextbook.com/physics/about.shtml, 2006.

[18] R. Felder and R. Brent. Understanding Student Differences. *Journal of Engineering Education*, 94(1):57–72, 2005.

[19] R. Felder and L. Silverman. Learning and Teaching Styles in Engineering Education. *Journal of Engineering Education*, 78(7):674–681, 1988.

[20] P. A. Fishwick. *Aesthetic Computing*. MIT Press, 2006.

[21] G. W. Furnas. Effective view navigation. In *CHI'97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 367–374. ACM Press, New York, 1997.

[22] P. A. Gloor. AACE - Algorithm Animation for Computer Science Education. *Proceedings of the 1992 IEEE Workshop on Visual Languages*, pages 25–31, 1992.

[23] R. Gomes, R. Seabra, J. Ventura, and N. Guedes. *SoundSpaces* – Behavior Based Computer Control System for Multimedia Systems. In *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE 2005), Monte de Caparica, Portugal*, page 407. ACM Press, New York, NY, 2005.

[24] O. Grillmeyer. An interactive multimedia textbook for introductory computer science. In *SIGCSE '99: The Proceedings of the thirtieth SIGCSE Technical Symposium on Computer Science Education*, pages 286–290, NewYork, NY, USA, 1999. ACM Press.

[25] S. Hansen, D. Schrimpsher, and N. H. Narayanan. From Algorithm Animations to Animation-Embedded Hypermedia Visualizations. *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA 1999), Seattle, Washington*, pages 1032–1037, 1999.

[26] M. Hitz and S. Kögeler. Teaching C++ on the WWW. In *Proceedings of the Annual Joint Conference on Integrating Technology into Computer Science Education (ITiCSE97), Uppsala, Sweden*, pages 11–13. ACM Press, New York, June 1997.

[27] C. D. Hundhausen. Integrating algorithm visualization technology into an undergraduate algorithms course: ethnographic studies of a social constructivist approach. *Computers & Education*, 39(3):237–260, 2002.

[28] C. D. Hundhausen and S. Douglas. SALSA and ALVIS: A Language and System for Constructing and Presenting Low Fidelity Algorithm Visualizations. *IEEE Symposium on Visual Languages, Los Alamitos, California*, pages 67–68, 2000.

[29] L. J. Issing and P. Klimsa, editors. *Information und Lernen mit Multimedia*. Psychologie Verlags Union, Weinheim, Basel, 2nd edition, 1997.

[30] D. H. Jonassen, T. Mayes, and R. McAleese. A Manifesto for a Constructivist Approach to Uses of Technology in Higher Education. In T.M. Duffy, J. Lowyck, and D.H. Jonassen, editors, *Designing Environments for Constructive Learning*, volume 105 of *NATO ASI, Series F, Computer and System Sciences*, pages 231–247. Springer, Berlin, Heidelberg, New York, London, 1993.

[31] A. Kerren. Generation as Method for Explorative Learning in Computer Science Education. In *Proceedings of the 9th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE '04)*, pages 77–81, Leeds, UK, 2004. ACM Press, New York.

[32] A. Kerren. Learning by Generation in Computer Science Education. *Journal of Computer Science and Technology (JCS&T)*, 4(2):84–90, 2004.

[33] S. Khuri. Designing Effective Algorithm Visualizations. *First International Program Visualization Workshop, Porvoo, Finland. University of Joensuu Press*, pages 1–12, Feb. 2001.

[34] R. A. Knuth and D. J. Cunningham. Tools for Constructivism. In T.M. Duffy, J. Lowyck, and D.H. Jonassen, editors, *Designing Environments for Constructive Learning*, volume 105 of *NATO ASI, Series F, Computer and System Sciences*, pages 163–188. Springer, Berlin, Heidelberg, New York, London, 1993.

[35] T. Lauer. Learner Interaction with Algorithm Visualizations: Viewing vs. Changing vs. Constructing. In *Proceedings of the 11th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2006), Bologna, Italy*, pages 202–206. ACM Press, New York, 2006.

[36] Learning Technology Standards Committee of the IEEE. Learning Object Metadata (LOM). Online: http://ltsc.ieee.org/doc/wg12/LOM-WD3.htm, (seen April 22, 2002).

[37] L. Malmi, V. Karavirta, A. Korhonen, J. Nikander, O. Seppälä, and P. Silvasti. Visual algorithm simulation exercise system with automatic assessment: TRAKLA2. *Informatics in Education*, 3(2):267-288, 2004.

[38] R. Martínez-Unanue, M. Paredes-Velasco, C. Pareja-Flores, J. Urquiza-Fuentes, and J. A. Velázquez-Iturbide. Electronic books for programming education: A review and future prospects. In *Proceedings of the 7th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education (ITiCSE '02)*, pages 34–38, Arhus, Denmark, 2002. ACM Press.

[39] A. Moreno, N. Myller, E. Sutinen, and M. Ben-Ari. Visualizing Programs with Jeliot 3. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI 2004), Gallipoli (Lecce), Italy*, pages 373–380. ACM Press, New York, May 2004.

[40] N. Myller. Automatic Prediction Question Generation during Program Visualization. In *Proceedings of the 4th Program Visualization Workshop (PVW 2006), Florence, Italy*, pages 89–93. University of Florence, 2006.

[41] T. Naps. JHAVÉ – Addressing the Need to Support Algorithm Visualization with Tools for Active Engagement. *IEEE Computer Graphics and Applications*, 25(6):49–55, Dec. 2005.

[42] T. Naps, M. McNally, and S. Grissom. Realizing XML Driven Algorithm Visualization. In *Proceedings of the 4th Program Visualization Workshop (PVW 2006), Florence, Italy*, pages 1–5. University of Florence, 2006.

[43] T. L. Naps. Incorporating Algorithm Visualization into Educational Theory: A Challenge for the Future. *Informatik/ Informatique, Special Issue on Visualization of Software*, pages 17–21, Apr. 2001.

[44] 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.

[45] R. Oechsle and T. Morth. Peer review of animations developed by students. In *Proceedings of the 4th Program Visualization Workshop (PVW 2006), Florence, Italy*, pages 39–43. University of Florence, 2006.

[46] S. Papert and I. Harel. *Constructionism*. Ablex Publishing Corporation, 1991.

[47] S. H. Rodger and T.W. Finley. *JFLAP: An Interactive Formal Languages and Automata Package*. Jones and Bartlett, 2006.

[48] R. J. Ross. CS 580 hypertextbooks: Next-generation learning resources. Available online at http://www.cs.montana.edu/˜ross/personal/cs580.htm, 2006.

[49] R. J. Ross. Snapshots of the theory of computing. Available online at http://www.cs.montana.edu/webworks/projects/ snapshots/, 2006.

[50] R. J. Ross and M. T. Grinder. Hypertextbooks: Animated, Active Learning, Comprehensive Teaching and Learning Resources for the Web. In S. Diehl, editor, *Software Visualization*, number 2269 in Lecture Notes in Computer Science, pages 269–284. Springer, 2002.

[51] G. Rößling and B. Freisleben. ANIMAL: A System for Supporting Multiple Roles in Algorithm Animation. *Journal of Visual Languages and Computing*, 13(2):341–354, 2002.

[52] G. Rößling, J. Trnková, U. Langendorf, K. Tillack, and M. Mühlhäuser. Learner-Centric Online Teaching for Non-Computer Science Students. In *Proceedings of ED-MEDIA 2002, Denver, Colorado*, pages 1645–1650. AACE Press, Charlottesville, VA, 2002.

[53] SCORM definition and Resources. Online: http://www.adlnet.org/scorm/downloads.cfm, (seen September 24, 2002).

[54] J. Sitthiworachart and M. Joy. Effective Peer Assessment for Learning Computer Programming. In *Proceedings of the 9th Annual Conference on the Innovation and Technology in Computer Science Education (ITiCSE 2004), Leeds, UK*, pages 122–126, 2004.

[55] L. Stern, H. Søndergaard, and L. Naish. A Strategy for Managing Content Complexity in Algorithm Animation. *Proceedings of the 4th Annual ACM SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education (ITiCSE'99),* Cracow, Poland, pages 127–130, Sept. 1999.

[56] J. Urquiza-Fuentes and J. A. Velázquez-Iturbide. Effortless construction and management of program animations on the web. In R. C. R. W. H. Lau, Q. Li and W. Liu, editors, *Advances in Web-Based Learning ICWL 2005*, pages 163–173. Springer-Verlag, LNCS 3583, 2005.

[57] B. Weidenmann. Multicodierung und Multimodalität im Lernprozeß. In L. J. Issing and P. Klimsa, editors, *Information und Lernen mit Multimedia*, pages 65–84. Psychologie Verlags Union, Weinheim, Basel, 2. edition, 1997.