Progress in Interactive Learning with a Digital Library in Computer Science

Edward A. Fox, N. Dwight Barnette, Clifford A. Shaffer, Lenwood Heath,
William Wake, Lucy T. Nowell, JAN Lee, Deborah Hix, and H. Rex Hartson
Department of Computer Science
Virginia Tech, USA
E-mail: fox@vt.edu

Abstract: We describe our prototype digital library in computer science based on publications from ACM and other publishers, and the integrated client and server software devised from local development efforts (i.e., the Envision and MARIAN systems) as well as other components (e.g., Hyper-G, KMS, Mosaic, WATERS). We explain changes to curriculum and new courseware developed with the aid of this digital library, as well as our preliminary evaluation.

1 Problems Addressed

Computer science students should be better prepared than they have been to work in the field of Information Technology. They often have poor writing skills, and cannot articulate clear convictions about ethical or professional matters. Many dislike reading, have little exposure to technical reports or articles in the research literature, are unfamiliar with key concepts of library and information science, and so are not prepared for life-long learning in their profession. They are encouraged to create programs from scratch, rather than learning key concepts of software reuse. Almost all incoming students lack elementary network usage skills. Due to the recognized importance of the Internet and World Wide Web (WWW), computer science students will be expected to be able to create WWW hyperlinked documents, navigate, and search the WWW for information. In order to fully take advantage of the WWW, graduates must be able to communicate and collaborate in a networked environment.

Computer science faculty face serious challenges in dealing with these problems. They must motivate students to develop: scholarly skills of effective writing, interest in reading about new developments, awareness to behave in an ethical and professional manner, and ability to search the literature for solutions (e.g., algorithms, data structures, interface techniques). Interactive computing shows promise to help, but developing and porting suitable courseware is complex. For example, on the Virginia Tech campus, with around fifteen thousand computers, there are a wide variety of hardware platforms and software packages, and different, often incompatible, technologies are available in student rooms, departmental laboratories, and faculty offices. Though a great deal of interactive multimedia courseware has been developed worldwide, little has been prepared for computer science students. Further, many of the early experiments in electronic publishing and digital libraries have focused on disciplines other than computing, like chemistry, material science, or medicine.

2 Approach and Outline

In connection with a variety of research projects, we have applied key technologies of electronic publishing, networked multimedia, distributed hypermedia, visualization, digital libraries, personalized learning, and educational technology to solve the problems of computer science education [Fox, 1994] [Fox & Abdulla, 1994] [Fox & Barnette, 1994].

One aim is to engage students in learning by making their educational experience more interactive. For example, finding algorithm animation software too hard to use or develop animations with, we embarked on specifying and implementing a system for algorithm visualization. Seeing that students prefer to work with computers rather than read a textbook, we began to develop a digital library from the computer science literature. Noting that paper-based distribution of information was wasteful of resources, inflexible, and unexciting, we embarked on "paperless" course delivery with LCD projection units attached to networked computers and increased faculty-student interaction through electronic mail and tutoring. Our first largely paperless course ran in Fall 1993; in Spring 1995 we have 4 courses that are completely paperless and more than 8 others moving in that direction, using the World-Wide Web (WWW) as "glue" to tie together the content and many tools in our environment.

In Section 3 we outline our work on algorithm visualization to increase interaction related to learning about data structures, and related efforts on an interface library. In Section 4 we describe our efforts to build digital library software to satisfy the needs of computer science students, faculty, and practioners. Section 5 discusses how we use our own Envision system in concert with a variety of other tools to provide access to computer science content. In Section 6 we describe the activities of our faculty and assistants in developing courseware. Section 7 gives preliminary findings from our evaluation studies, and Section 8 summarizes our work.

3 Software for Interaction

When a student learns programming, he often finds it difficult to understand data structures and how they change. This applies to programs given to him and to those he writes himself. We wish to support visualiza- tion of the data structures in both kinds of programs for use in our courses in data structures and algorithms. Unfortunately, existing algorithm animation systems are much too difficult for students (and most instructors) to learn how to animate an algorithm. Thus, we have developed our own visualization system, Swan.

3.1 Algorithm Visualization with Swan

Using Swan, a C or C++ program can be annotated to provide views of the program's execution. The Swan annotation library is designed primarily for ease of use, so that instructors and students can annotate existing programs with relatively little effort, much as a programmer might place print statements in a program to get information to help in debugging. Swan is designed to support visualization of many types of graphs.

Our goal is to provide the viewer with a better understanding of a program's data structures and execution. One innovative feature of Swan is that it allows visualization to be a two-way communication process between the view and the program. In particular, the algorithm animator can provide the viewer with the ability to modify data structures as they proceed through a visualization. In this way, students can experiment with a data structure or an algorithm to gain further understanding of its operation.

Several graph layout algorithms are implemented in Swan, including special ones for arrays, lists and trees. Multiple algorithms are available for automatic layout of general graphs. Automatic layout allows the annotator to concentrate on the logical structures of views without worrying about their graphical display. Swan annotation functions also allow the annotator to give precise specifications for graph layout, should this be desired. Thus, the annotator has a range of options in terms of how much effort to put into graph layout.

Swan has been designed to serve many roles: as a presentation medium for instruction in data structures and algorithms courses, as a graphical debugger for students in programming courses, and as a platform for experimenting with various graph layout algorithms.

3.2 The GeoSim Interface Library for Introductory Programming Courses

The GeoSim Interface Library (GIL) [Shaffer, 1994] provides a common set of Graphical User Interface functions for programs running under the MS-DOS, Macintosh and X Windows programming environments. It is easy to learn and use, and allows student programmers to quickly develop GUI interfaces for their programs. Beginning in Spring 1996, second semester programming students will begin using the GIL library as a part of their regular programming assignments. In this way, students will be given early exposure to programming GUI interfaces, an experience usually not available except through senior elective courses. GIL, including sourcecode and documentation, will be freely distributed via the Internet.

4 Digital Library Software Development

The Envision Project [Fox et al., 1993b] [Heath et al., 1995] was launched in Fall 1991 to follow a user-centered development process for a digital library in computer science. We began with user interviews and task analysis, embarked upon system and interface design, and started looking for suitable tools. A system prototype was constructed on a NeXTstation, and demonstrated the feasibility of our early design. However, for portability reasons, we limited subsequent use of NeXT software to the search engine component of our system, MARIAN [Fox et al., 1993a].

In particular, we adapted the MARIAN online public access retrieval system already under development to handle searching for Envision. This required layering around MARIAN with C++ code developed on DEC Alpha systems under OSF/1, for increased performance, handling of SGML documents, and construction of a simple but tailored object-oriented DBMS.

The Envision server maintains multiple query sessions, each a thread that communicates with one Envision client and that is the intermediary between the session and the MARIAN searchers. The server packages a query in the form of a remote procedure call to MARIAN and receives a ranked set of document identifiers (IDs) in response. For each document ID, the server constructs a profile that is returned to the Envision client, including author, title, year, type and publication. The Envision client utilizes this profile information to create our innovative visualization of result sets [Nowell et al., 1994].

Documents that are entered into the Envision library must be represented by an SGML document, preferably the entire text but at least a bibliographic record and abstract. The SGML document is the source of the indexing information used by MARIAN, and, in addition, can be rendered using our SGML-to-HTML translator.

It is not feasible by hand to convert to SGML large numbers of documents created in a variety of formats. Recognizing the need to automate this process, we developed DELTO, an analysis and translation system capable of extracting structural information from a document, analyzing its components, and producing an SGML version. DELTO also extracts hypertext links to other documents, authors, institutions, and objects.

We chose X/Motif as our windowing system, since it was widely used, including by our faculty and students (who are required to buy UNIX workstations as freshmen). Therefore we began our interface development in that environment [Nowell et al., 1994].

5 Digital Library Systems and Content

ACM has been supplying content in a variety of forms since 1991 to be used with our Envision system. We developed document type definitions (DTDs) for bibliographic records and for full-text articles. All new articles in the various *Transactions* have been converted semi-automatically to an SGML encoding. Portions of *CACM* articles have been similarly converted. Bibliographic data from *Guide to the Computing Literature*, the *HCI Biliography*, and various other sources have been marked up using SGML.

Separately, we scanned in thousands of page images, using CCITT Group 4 FAX compression into TIFF files at 300 dpi. Each such article has a corresponding bibliographic record and abstract (corrected from the OCR version). These cover many of the readings that faculty have indicated are most relevant to their courses.

ACM has also provided *DA Library* and CD-ROM versions of various conference proceedings (e.g., SIGGRAPH, Multimedia), with page images or Adobe PDF files. We have two Pioneer changers each able to provide access to 6 CD-ROMs. Late in 1995 we expect a large number of CD-ROMs containing page images of articles from IEEE-CS and will make those accessible too.

In addition to content targeted for the Envision system, we have a variety of other systems containing important information [Fox & Barnette, 1994]. Knowledge Systems Inc. has provided its KMS hypertext system [Akscyn et al., 1988], running on our SPARC-10, and ported it to DECstations for our students. KMS has been useful for teaching about hypertext and computer-supported cooperative work, and allows handy access to the *ACM Hypertext Compendium.*

For computer science technical reports we have been involved in development of the WATERS system [Maly et al., 1994] [French et al., 1995]. Almost all of our department's reports have been scanned in as TIFF images or made available as compressed PostScript files. An April workshop at NSF Headquarters will explore development and interoperability issues to cover the entire set of reports prepared nationally.

The Hyper-G system, also running on our SPARC-10, provides access to the *Journal of Universal Computer Science.* In addition, we arranged with the editor of the internet digest *RISKS* to keep an up-to-date fully indexed and browsable (by volume, issue, date) version accessible on the WWW through Hyper-G. This process will be repeated for *ReNews* digest too. Further use of Hyper-G may provide hypermedia access to particular portions of images and read-only data, since anchors at such locations cannot be represented easily with other WWW tools. Adding Hyper-G to the WWW enriches our courseware development environment.

6 Courseware Development

In parallel with digital library construction, faculty, graduate assistants, and several undergraduates engaged in special research projects have worked extensively on courseware. Please visit http://ei.cs.vt.edu/courses.html to examine our evolving collection.

First efforts related to the fall course, CS5604, *Information Storage and Retrieval,* taken by graduate students interested in text processing, electronic publishing, searching, and related research. In 1993 the course was revised to use gopher and WWW in a move toward paperless delivery. At the same time the *Keller Plan,* or Personalized System of Instruction, was adapted, with the course divided into units. In 1994 the course was changed to require no paper whatsoever, delivered instead on the WWW and systems discussed in Section 5.

Also in Fall 1994 the junior level course on *Computer Professionalism* was revamped to make extensive use of writing, through an adaptation of the internet tool *WIT.* At the same time a new 1-credit "information literacy" course, now assigned the number CS1604, *Introduction to Networked Information,* was offered to about 40 students. The enrollment is expected to double at the next offerings: Spring 1995, Summer, and Fall. In addition to being run at Virginia Tech, a version is being given at Norfolk State University, and funding has been assigned as part of the SUCCEED coalition to use distance learning methods, interactive digital video, and other methods to offer this course at many other locations. This paperless course will have both a one-hour lecture and a self-study version, allowing comparisons between these two approaches. The course focuses upon introducing students to knowledge needed to produce and locate hyperlinked information on the WWW. Commercial and noncommercial networks are introduced also as examples. Assignments (submitted through email), deal with using various WWW tools (Mosaic, Gopher, Archie, WAIS, etc.), to find, retrieve and post WWW information.

Another new course, first offered in Spring 1995, is *Multimedia, Hypertext and Information Access.* This senior level course runs in a paperless fashion, follows the *Keller Plan,* and takes place in a laboratory of Power Macintosh 6100/60 AV computers. Students are quite comfortable accessing materials prepared for the CS5604 and CS1604 classes, despite their being at different levels, as well as a variety of related information available over the WWW. It is clear that a rich pool of courseware is growing on the Internet, with instructors starting to share and collaborate on an unprecedented level.

The data structures course is moving toward on-line distribution of assignments and materials. Students often contact the instructor by email. Online copies of all assignments are available, using a standard LaTeX to HTML converter. A copy of the overhead transparancies used during lecture are available online. All grades are posted online, rather than outside the instructors door. An ongoing question and answer series for clarification of programming assignments is maintained electronically. Next year, we hope to provide a series of data structure visualizations based on Swan, available for downloading by the students through WWW.

Besides these courses, other efforts are underway or planned in areas including Algorithms, Computer Literacy, Human-Computer Interaction, Modeling and Simulation, and Numerical Analysis. We welcome comments and suggestions on our work and plans, and hope that our courseware will be of interest to others.

7 Evaluation

Evaluating a large effort like this is quite complex. Our courseware server, ei.cs.vt.edu, is accessed about 20K times per week. A Ph.D. dissertation is underway to evaluate the Envision system. Another Ph.D. dissertation is proceeding considering browsing methods relating to Envision and MARIAN. A third Ph.D. dissertation is starting, dealing with modeling and simulation studies relating to our projects.

For the last two years, student ratings of CS5604 have improved considerably, even though the course has been taught by the same instructor for over a decade. In all the course offerings involving the Keller Plan, the students have been enthusiastic about this approach. Students who took CS5604 earlier have found it easy to review materials online, too.

The first offering of CS1604, *Introduction to Networked Information,* in the Fall of 1995, yielded expected results from the pre-test and post-test. The tests were designed to determine the standard computer literacy background of the students, along with their experience and knowledge of networked hypermedia information. The pre-test confirmed that the students' computer literacy background was limited primarily to standard business applications (word processing, spreadsheets and data base systems). Very few subjects had prior experience with the internet or WWW, and almost none had used common network applications (FTP, Mosaic, Gopher, Archie, WAIS, etc.). The majority of students also had not been previously exposed to digital library applications (OPACs, Electronic Journals). The post-test verified positive effects of the student exposure to the course. All of the aforementioned survey results showed marked improvements. While no truly conclusive results can be drawn at this point, the initial results demonstrate great promise. Additional questions were included on the post-test aimed at determining student reaction to a hypermedia networked paperless course of this type. These questions yielded favorable responses. The goal of a long term effect upon the students' learning will be undertaken by a followup survey that is to be administered each of the following 3 years after students have completed the course.

We have begun annual collection of demographic data department-wide, to help identify noticeable trends. Each course using our digital library has a pre-test and post-test administered. Accesses to courseware pages are logged, as people from around the world are referring to these and other course materials we have prepared.

8 Summary

We have worked to develop a digital library in computer science and to revamp our courses to benefit therefrom. A variety of systems have been employed, and with the WWW as "glue", fit neatly into the educational experience of our students. Our new paperless courses are well regarded by those who take them, who seem particularly motivated and are therefore willing to work and learn more.

9 References

[Akscyn et al., 1988] Akscyn, R., McCracken, D., & Yoder, E. (1988). KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations. *Commun. of the ACM,* 31(7).

[Fox, 1994] Fox, E. (1994). How to make intelligent digital libraries. *Methodologies for Intelligent Systems, Proc. 8th International Symposium, ISMIS '94,* Charlotte, NC, Oct. 1994. Lecture Notes in Artificial Intelligence 869, Springer-Verlag, Berlin. 27-38.

[Fox & Abdulla, 1994] Fox, E. & Abdulla, G. (1994). Digital Video Delivery for a Digital Library in Computer Science. *High-Speed Networking and Multimedia Computing Workshop, IS&T/SPIE Symposium on Electronic Imaging Science and Technology.* Feb. 6-10, 1994, San Jose, CA, 7 pages.

[Fox & Barnette, 1994] Fox, E. & Barnette, D. (1994). Improving Education through a Computer Science Digital Library with Three Types of WWW Servers. *Proc. Second International WWW '94: Mosaic and the Web, WWW'94,* Chicago, IL, Oct. 17-20, 1994.

[Fox et al., 1993a] Fox, E., France, R., Sahle, E., Daoud, A., & Cline, B. (1993). Development of a Modern OPAC: From REVTOLC to MARIAN. *Proc. 16th Annual Intern'l ACM SIGIR Conf. on R & D in Information Retrieval, SIGIR '93,* Pittsburgh, PA, June 27 - July 1, 1993, 248-259.

[Fox et al., 1993b] Fox, E., Hix, D., Nowell, L., Brueni, D., Wake, W., Heath, L., & Rao, D. (1993). Users, User Interfaces, and Objects: Envision, a Digital Library. *J. of the American Society for Information Science,* Sept. 1993, 44(8): 480-491.

[French et al., 1995] French, J., Fox, E., Maly, K., & Selman, A. (1995). Wide Area Technical Report Service --- technical reports online. *Commun. of the ACM,* Apr.1995, 38(4), 1 page, in press.

[Heath et al., 1995] Heath, L., Hix, D., Nowell, L., Wake, W., Averboch, G., Labow, E., Guyer, S., Brueni, D., Fox, E. (1995). Envision: A User-Centered Database of Computer Science Literature. *Commun. of the ACM,* Apr. 1995, 38(4), 1 page, in press.

[Maly et al., 1994] Maly, K., French, J., Selman, A. & Fox, E. (1994). The Wide Area Technical Report Server. *Proc. Second International WWW '94: Mosaic and the Web, WWW'94,* Chicago, IL, Oct. 17-20, 1994.

[Shaffer, 1994] Shaffer, C. (1994). *The GeoSim Interface Library (GIL): Programmer's Manual.* Blacksburg, VA: VPI&SU Dept. of Computer Science TR 94-31.

[Nowell et al., 1994] Nowell, L., Fox, E., Heath, L., Hix, D., Wake, W., & Labow, E. (1994). *Seeing Things Your Way: Information Visualization for a User-Centered Database of Computer Science Literature.* Blacksburg, VA: VPI&SU Dept. of Computer Science TR 94-06.

Acknowledgments

We thank all of those funding this work, especially the National Science Foundation. ACM and IEEE-CS have been generous in allowing access to their materials. KSI has provided KMS, and Technical University of Graz has provided Hyper-G. Cost sharing has been given by Virginia Tech's Department of Computer Science, College of Arts and Sciences and Computing Center.

Sandra DeLoatch and John Urquhart at Norfolk State University have collaborated as co-PIs. Other investigators at Virginia Tech include Marc Abrams, Osman Balci, and Judy Watson. Staff involved include Ben Cline and Robert France. Students helping have included Guillermo Averboch, Dennis Brueni, Scott Guyer, Chris Kirmse, Eric Labow, and Steve Teske.